Language identification in document images

P. Barlas, D. Hebert, C. Chatelain, S. Adam, T. Paquet
Universite de Rouen & INSA de Rouen, LITIS EA 4108, BP12, 76801 Saint Etienne du Rouvray, FRANCE

Abstract

This paper presents a system dedicated to automatic language identification of text regions in heterogeneous and complex documents. This system is able to process documents with mixed printed and handwritten text and various layouts. To handle such a problem, we propose a system that performs the following sub-tasks: writing type identification (printed/handwritten), script identification and language identification. The methods for the writing type recognition and the script discrimination are based on the analysis of the connected components while the language identification approach relies on a statistical text analysis, which requires a recognition engine. We evaluate the system on a new public dataset and present detailed results on the three tasks. Our system outperforms the Google plug-in evaluated on the ground-truth transcriptions of the same dataset.

1. Introduction

Identifying the language(s) of a document is a key step of a document reading system since recognition engines require the integration of a language model to increase the transcription performance. In this article, we address this task in a very difficult context where documents are unconstrained, mix variable writing types (handwritten and printed) and two different scripts/alphabets (Latin and Arabic). To the best of our knowledge, this challenge has never been handled in the literature.

The proposed approach for identifying the language of a document image, already introduced in [12], rely on a sequential system illustrated in Figure 1. First, text blocks are extracted by a segmentation stage described in [9]. Then, the writing type (handwritten vs. printed) of each text block is identified through an analysis of the connected components using codebooks of contour fragments. A similar approach is then used to identify the script. This second stage takes advantage of the writing type information to choose an optimal codebook configuration. If an Arabic script is decided, the block language is considered to be Arabic. For Latin block, the language identification is performed by exploiting the outputs of a recognition engine. A statistical analysis is carried out analyzing separately the transcription of printed blocks and handwritten blocks.

The overall system is evaluated on the new publicly available MAURDOR dataset [6]. This dataset contains heterogeneous documents (forms, printed and manually annotated business documents, handwritten correspondence, maps, ID, newspapers articles, blue-prints, etc.), with mixed printed and handwritten texts, in various languages (French, English and Arabic). The MAURDOR dataset represents a challenge for numerous tasks in the domain of document image analysis : namely document layout analysis, writing type identification, language identification, text recognition and semantic information extraction (reading order, dates, address blocks, etc.). The results obtained on the tasks of writing type and script identification compare favorably with the state of the art. Moreover, our language identification system outperforms the Google plug-in [14] which has been evaluated on the ground-truth transcriptions of the MAURDOR dataset.

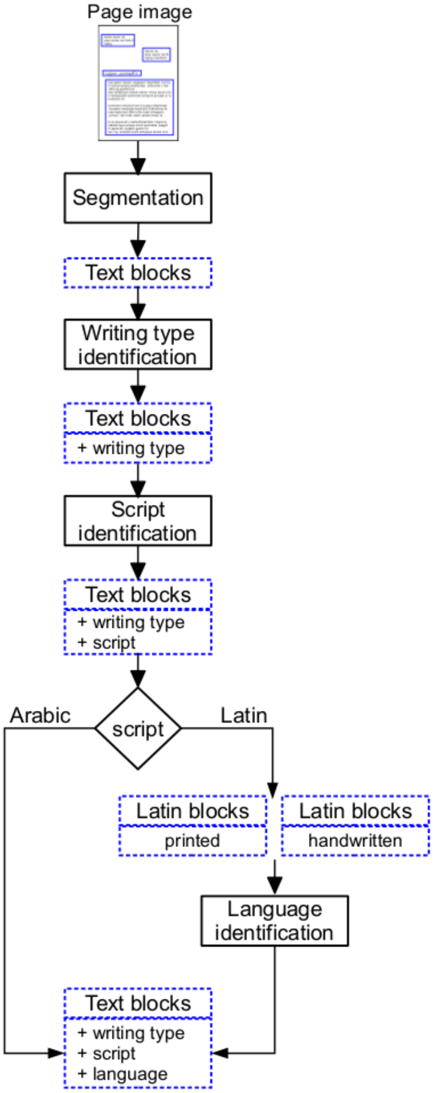

Figure 1. *The proposed approach for writing type, script and language identification*

This paper is organized as follows. In the next section a complete literature review of the works dedicated to language identification as well as script and writing type identification is presented. Then, the writing type and script approaches are described before detailing the language identification approach. The following section presents a detailed analysis of the experimental results obtained on the documents of the MAURDOR dataset. Finally, the paper concludes with a brief summary and a discussion of future works.

2. Related Works

Language identification can be considered in two scopes of application: electronic documents and document images. On electronic documents, language identification is now considered as a solved problem. Reliable systems with high accuracy are available. As an example, the Google plug-in described in [14] reaches a precision over 99% for 53 languages using n-gram of characters and language profiles. On the contrary, language identification is still a challenging issue on document images. The works handling this problem are rare [22, 23, 24] and are focused on machine-printed writing. To the best of our knowledge, the only approach dedicated to language identification on handwritten documents images [25] is also based on shape features.

When working on unconstrained documents mixing printed and handwritten text in languages with different scripts, the writing type and the script are relevant information that need to be detected prior to language identification. The literature for these two steps is abundant for printed documents, but less for handwritten documents. Table 1 proposes a synthesis of the literature for language, script and writing type identification. In the following, we review the methodologies involved for each of these tasks.

Language identification: Most of the works devoted to language identification are designed to deal with electronic documents, where the text is directly available [14, 18, 20, 19, 21, 15, 16, 17]. These approaches rely on language models and statistical analysis of characters [18], or on the detection of keywords/short words [19] or n-grams of characters [14, 20, 21, 19]. [15] made a combination of these three types of analysis with a ranking combination strategy to improve the identification rate on two electronic document databases. Also based on n-grams, [18] relies on Markov models to model each language and tries to find the best fitting model for a new sequence of characters. More recently, [20] has defined a n-gram method able to identify the language on short texts of same language and on texts composed of multiple languages. [16] combines n-grams with heuristics and the Lin's similarity measure to identify 12 languages (Danish, English, Italian, Spanish, French ...). [17] proposes a graph-based n-gram approach for its system called LIGA to identify the language on short and ill-written texts (Twitter messages).

As said before, only few methods are dedicated to language identification on document images [22, 23, 24, 25] and, in most cases, language identification is performed on printed documents without any OCR. Both [22, 23] apply shape coding approaches. [22] creates character shape codes gathering family of characters (e.g. one code represents all the characters with ascenders) whereas [23] builds word shape codes based on character extremum points and the number of horizontal word runs. Once shape codes extracted, [23] measures the similarity between the language templates and the document vector. In [24], English and German languages are identified using language models. A general model (gathering the most frequent words unigram in the five Latin languages) is first generated applying a Latin OCR on the documents of a training set. This general model is used to generate each language model measuring the number of occurrences of each word of the general models in the training set of the language. The language identification is then performed computing the word unigram relative entropy for each language. Regarding the language identification on handwritten documents, [25] proposes an approach based on the shape analysis of the connected components of the handwritten document to discriminate the script (Arabic, Cyrillic, Devanagari, Japanese, Latin) and the language (English, German). A document is characterized by the means, the standard deviation and the skew of five features encoding connected components properties (aspect ratio, compactness, number of holes, centroid positions). The classification is performed using a linear discriminant analysis and the system was tested on a private database composed of cleaned images (the irregularities are removed after scanning).

This review of the literature devoted to language identification has shown that works were mainly focused on digital documents. These approaches are based on statistical text analysis or on the detection of keywords or n-grams and all achieve high performance with an average accuracy classification around 99%. On the other hand, approaches dedicated to language identification on document images are very few and the problem is more complicated given that text information is not available. Existing approaches in the literature are focused on printed documents. They work at the document level and use mainly shape analysis. Both [22, 23] reach an average accuracy above 90% using shape coding approaches considering respectively 23 and 8 languages, whereas [24], combining spatial features with the analysis of OCR outputs, achieves an average precision of 94.76% on a private dataset composed of fax images, considering 7 languages. The only approach dedicated to handwritten documents [25] achieves a classification average accuracy around 85% for the discrimination of german/english languages, on images previously cleaned with Adobe Photoshop in order to remove any irregularities (illustrations, doodles, anomalous writing, etc.).

Script identification: In some cases, the identification of a language can be performed directly by detecting its script (e.g. Arabic). As a consequence, language identification should be coupled with script identification approaches. The majority of recent works devoted to script identification consider printed documents [30, 26, 27, 31, 29, 32, 23]. Only few recent works handle both printed and handwritten documents [34, 35, 28, 33]. The methods working at the document level are based on shape analysis. [26, 27] use respectively bounding boxes distributions and average pixels distributions. [23] generates script templates using a clustering approach based on the distribution of vertical runs. Among the methods working on text zones or word images, some works use similar approaches. [30, 26] use profile analysis on connected components or on images of lines and words. [31] builds a template extracting Arabic character segments in order to separate Arabic words and Latin words. [29, 32] use texture based approaches on printed documents. The images are filtered with gabor filters and steerable gabor filters and the mean and standard deviation of the filtered images are extracted to feed a classifier (MLP/KNN). [33] performs script identification on printed and handwritten documents covering 8 scripts (Arabic, Chinese, En-

Table 1: Writing type, script and language recognition methods

	Problem						Method	
ref.	problem	database	scri./lang.	scope of application			features	decision
[14]	language	Wikipedia	**53 lang.**	digital	-	line	n-gram of characters	Naive Bayes
[15]	language	Leipzig Corpora Collection and Wikipedia	**13 lang.**	digital	-	doc.	combi. of short, freq. words & n-gram	Ad-Hoc Ranking
[16]	language	Web pages	**12 lang.**	digital	-	doc.	n-gram + heurist.	similarity measu.
[17]	language	Twitter messages	**6 lang.** Ger.,Eng.	digital	-	paragraph	graph of 3-gram order and frequencies over languages	path matching score
[18]	language	private database.	Spanish, English	digital	-	line, paragraph	characters (context with the order of the Markov model)	Markov models of various order with baye. deci. rule
[19]	language	sentences database from ECI CD-Rom	**9 langu.**	digital	-	sent.	3-grams of characters or short words	normalized frequ. comparison
[20]	language	Wikipedia	**8 langu.**	digital	-	word, line	n-gram + dictionary	
[21]	language	Usenet newsgroups	**8 langu.**	digital	-	word	n-gram of characters on tokens (2, 3 and 4-gram)	Ad-Hoc Ranking
[22]	language	private database	**23 lang.**	image	print.	doc.	character shape codes	LDA model
[23]	language	private database	**8 langu.**	image	print.	doc.	doc. vectorization based on word shape codes	simil. between doc. vector and lang. templates
[24]	language + script	private database	**7 langu.** Asian/Lat. scripts	image	print.	doc.	spatial features + language models based on OCR outputs	word unigram relative entropy
[25]	script + language	private database	**6 script.** Eng./Ger.	image	hand.	doc.	physical (CC aspect ratio, centroid pos., compactness, etc.)	LDA
[23]	script	private database	**8 scripts** Ar., Lat., Chin.,...	image	print.	doc.	generation of templates using clustering (density and distrib. of vert. runs)	Bray Curtis distance to the script templates
[26]	script	private database (business let., newspapers, flyers,...)	Latin, Arabic, Ideogra.	image	print.	doc.	physical (bounding boxes distrib., hor. proj.)	rules based classification
[27]	script	private database (magazines, newspapers, etc.)	Kan.,Hin., Urd.,Eng.	image	print.	doc.	physical (average pixels distrib. after morph. op.)	KNN
[28]	script	private database (postal images)	Bangla, English	image	print. hand.	zone	physical (CC profiles analysis)	3 rules system
[29]	script	private database (magazines, books, etc.)	Chi.,Jap., Kor.,Eng.	image	print.	zone	texture (steerable gabor filter)	MLP
[30]	script	private database	Arabic, English	image	print.	line, word	physical (proj. profile analysis, runlength hitso.)	MLP
[31]	script	private database (scientific articles)	Arabic, Latin	image	print.	word	Arabic character segments	template matching
[32]	script	private database	Ar.,Hin., Kor.,Eng., Chi.	image	print.	word	texture (gabor filter)	KNN
[33]	script	University of Maryland database + IAM-DB	**8 scripts** Ar.,Chi., Eng.,...	image	print. hand.	doc.	codebook of generic shape features (modified kAS)	SVM
[34]	print./hand. + script	private database	Arabic, Latin	image	print. hand.	zone	physical (block: nb of diacritics, occlusions, CC: density, eccentri., etc.)	KNN
[35]	print./hand. + script	IAM-DB, IFNENIT, words from magazines & newspapers for print.	Arabic, Latin	image	print. hand.	word	features of the literature (vert. proj. var., CC width/height, etc.)	compar. : Bayes, KNN, SVM, MLP
[36]	print./hand.	private database	Arabic	image	-	zone	physical (codebook of TAS)	SVM
[37]	print./hand.	IAM-DB, GRUHD	English, Greek	image	-	zone, line	physical (upper and lower horizontal profile)	discriminant analysis (ANOVA)
[38]	print./hand.	private database (magazines, newspapers, handmade images)	English, Chinese	image	-	zone, line	spatial (character blocks layout)	threshold on the block layout variance
[39]	print./hand.	private database (business letters)	English	image	-	zone, word	physical (region size, density, CC var., etc.)	Fisher classifiers
[40]	print./hand.	MAURDOR database	English, French, Arabic	image	-	zone, word	physical (width, height, surface, Zernike moments, etc..)	Boosting bonsai trees
[41]	print./hand.	IAM-DB	English	image	-	word	physical (CC area, perim., compact., etc.)	KNN
[42]	print./hand.	private database	English	image	-	word	physical (proj. profiles)	HMM
[43]	print./hand.	Nist database (hand.) & private database (print.)	Latin	image	-	char.	physical (straightness of vert./hor. lines)	MLP
[44]	print./hand.	ETL character database	Chinese	image	-	char.	frequency (fluctuations caused by handwriting)	MLP

glish, Hindi, Japanese, Korean, Russian and Thai). A shape codebook is first constructed by clustering shape codewords based on k-Adjacent Segments (kAS). The image of document is characterized by the occurrences of codewords of the shape codebook in the image. Finally, a multi-class SVM is used to detect the script.

Some other methods were interested in both writing type and script identification (Arabic/Latin). [34] performs a zone classification using a KNN and physical features extracted at both level: the block level (number of occlusions, diacritics . . .) and the connected component level (density, eccentricity . . .). [35] performs a feature selection among the features proposed in the literature (projection profile, connected components with/height, steerable pyramid . . .) and compares different classifiers and achieves best performance with a Bayes classifier.

The approaches of the literature for script identification are generally based on shape or texture analysis coupled with classifiers. These approaches achieve an average classification accuracy within a range from 91% in [26] to 99.7% in [30], all using printed documents and private datasets. [30] achieves high performance testing the approach on text lines extracted from Arabic and English magazines. Approaches of the literature working on both printed and handwritten text are very few. Both [28, 33] approaches are using shape analysis and reach an average accuracy around 95% on a private dataset composed of postal images [28] and on the IAM-DB and the University of Maryland datasets [33]. Two other methods [34, 35] perform script identification as well as writing type discrimination. Also based on shape analysis combined with classifiers, these approaches achieve a global rate classification within a range from 88% in [34] to 98.72% in [35]. The latter reaches high performance experimenting the approach using one different datasets for each class.

Writing type identification: Language identification on documents mixing printed and handwritten text requires to proceed to the writing type identification when the information is not available. A majority of methods focuses on Latin documents and more precisely on English documents, but some recent works are dedicated to Arabic [36, 40], Chinese [44, 38] and Greek documents [37]. The methods working at the zone or word level can be grouped regarding the features used. [39, 41] base their approaches on the analysis of physical descriptors of the regions (size, density . . .) as well as on the connected components (area, size, variance . . .). In [40], the authors use region size features, as well as center and moments of inertia, Zernike moments and histogram of Freeman directions, making a 244-dimensional features vector. Features are then selected using the bonzaiboost system based on the Adaboost algorithm combined with small decision trees. In [37, 42], the authors use the regularity of the printed writing, extracting upper and lower horizontal profiles to estimate the stability of the printed characters [37], or using an algorithm based on Hidden Markov Models (HMM) to measure the regularity of the projection profile [42]. [36] is interested in printed/handwritten writing classification in Arabic documents. The approach relies on a SVM classifier fed with shape based features using codebooks of Triple Adjacent Segments (TAS). Another possible approach when working at the zone level is to use spatial features. [38] analyzes the layout of characters in the block applied to either English or Chinese documents. The methods working at character level analyze the regularity of the writing. [43] analyzes the straightness and the symmetry of Latin printed characters whereas [44] bases its approach on the fluctuations caused by the handwriting, transforming Chinese characters into the frequency domain. Both approaches use neural networks to take the decision.

The review of the literature has shown that writing type identification in Latin documents is widely covered by the existing approaches. These methods are all based on the shape analysis of the document and obtain an average accuracy ranging from 85% in [38] to 98.57% in [41]. Among the approaches working on Latin documents and achieving an accuracy rate around 98% [41, 39, 37], two were evaluated on the IAM-DB dataset and the other on a private dataset composed of business letters. At present and to the best of our knowledge, two works [36, 40] handle Arabic documents. The first one reaches a pixel-weighted zone classification accuracy of 98% using a codebook based approach on an Arabic private dataset. The second approach obtains an average classification accuracy of 91.1% and 94.07% (depending on the system configuration) for the writing type identification on Arabic and Latin documents of the MAURDOR dataset. Regarding the approaches working at the character level, [43] achieves an accuracy of 78.5% on the NIST dataset for handwritten characters and on a private dataset for printed characters.

One can see in Table 1 that script and writing type identification are based on similar techniques based on shape analysis and a classification stage. A couple of approaches combine script identification with writing type detection [34, 35]. However, script identification methods are mainly dedicated to printed documents. Moreover, Table 1 highlights the fact that only a few works of the literature perform language identification on printed document images and approaches working on handwritten document images are even more rare. However, real-life documents tend to mix handwritten and printed writings (annotations, application forms, medical receipts, . . .). Applying an OCR on such documents is still a challenging issue. It requires to separate handwritten text blocks from printed blocks as well as identifying the language of the document in order to select the appropriate configuration for the OCR. Figure 2 shows some examples of text blocks illustrating the difficulties of the problem. First of all, we can notice that the amount of information in text blocks can be heterogeneous. A text block can be composed of a single character up to several paragraphs. Consequently, the systems need to face the variability of the block contents to take a decision. We can also notice that the script discrimination (Arabic/Latin) on printed documents can be made by shape analysis of the blocks since the different scripts are of different nature (cursive style and printscript style). However, the problem becomes more difficult on handwritten documents since the handwriting can be both printscript and cursive styles. Finally, the use of shape analysis for languages sharing the same alphabet (French/English) seems to be limited and a textual analysis using an OCR approach would be more suitable.

In this article, we propose a method for language identification on document images mixing printed and handwritten texts for three different languages (French, English and Arabic). Our language identification system is able to tackle the three sub-tasks: writing type identification, script identification and language identification. Including writing type identification in our system enables us to handle any kind of document without the need of knowing the type of document, or any other information required for the recognition stage. Writing type and script identification methods are based on a same approach using a codebook-based feature set. The approach for language identification relies on the statistical analysis of a Latin OCR output.

3. Writing type and script identification system

Before the task of language identification, the writing type and the script of text blocks need to be identified. These two tasks are handled with the same approach with different configurations. The approach proposed for writing type and script identification is based on the shape analysis of connected components and therefore does not require any recognition stage.

Writing type and script identification are performed on text blocks that may contain either several paragraphs, only few words, or even a single character. The content of a text block being variable, we use a decision at the connected component level so as to determine the writing type or the script of the text block. As a consequence, connected components are extracted from the block and the classification of each component is performed using a codebook based approach, inspired from writer identification methods described in [3] and [4]. The classification of a connected component is performed through the extraction of its contour fragments. These local shape descriptors enable us to encode small fragments of characters which are efficient features for the writing type separation especially when a printed script is cursive such as the Arabic script. Moreover, methods using local shape descriptors are more efficient than methods using spatial information [38] when there is less content in a text block. The contour fragments of the connected components are compared with fragments of a codebook and a bag of contour fragments is used as a feature vector to classify the connected components using a MLP classifier. The final step consists in identifying the writing type or the script of a text area using a majority vote on the decisions taken for each of the connected components of the text block.

In the following sections we detail the important steps of our approach and the applications for writing type and script identification.

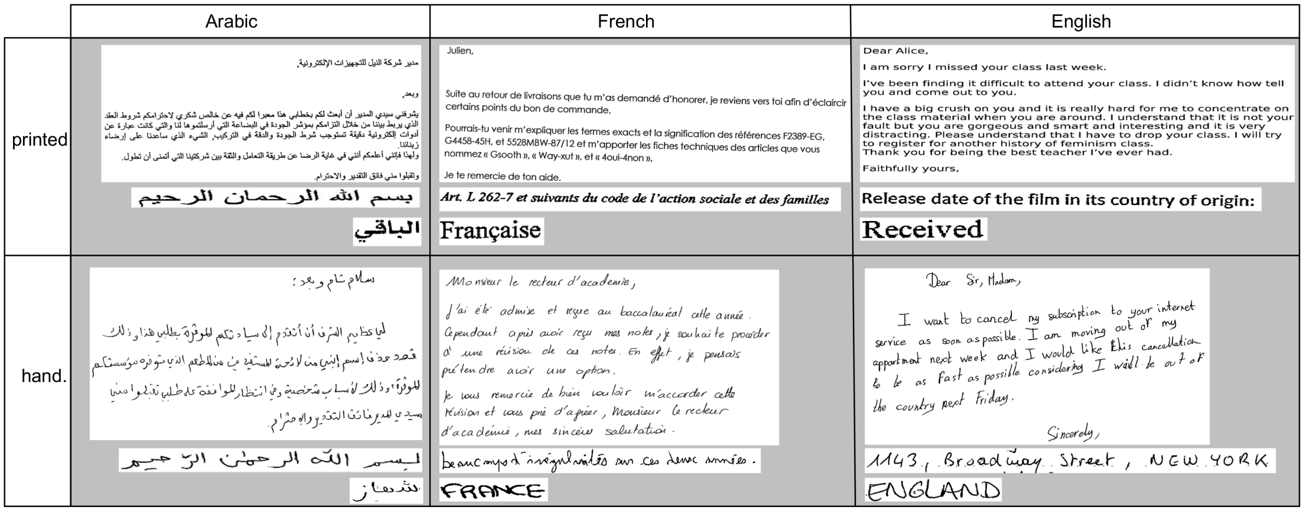
Arabic
French
English
printed
hand.
بسم الله الرحمان الرحيم
الباقي
Art. L 262-7 et suivants du code de l'action sociale et des familles
Française
Release date of the film in its country of origin:
Received
بسم الله الرحمن الرحيم
beaucoup d'irrégularités sur ces deux années.
FRANCE
1143, Broadway Street, NEW YORK
ENGLAND

Figure 2. Examples of text blocks for all writing types and languages in the MAURDOR dataset: they can be composed of paragraphs, or more often only few words.

Contour fragment based approach

Fragment extraction and representation

Fragmented parts of writings differ according to their writing type or their script. An efficient way to capture local shape properties of a writing is to extract fragments of the external contour of its connected components. A contour fragment is defined by its length l and an overlap of fixed size s between two adjacent fragments, moving along the external contour of the connected component as illustrated on Figure 3. The overlap represents the number of pixels shared by fragment i and fragment $i+1$. Fragments are extracted over the whole contour of the connected component, without any normalization. We choose to represent fragments using the ChainCode Histogram (CCH) described in [5] which is a translation and scale invariant shape measure.

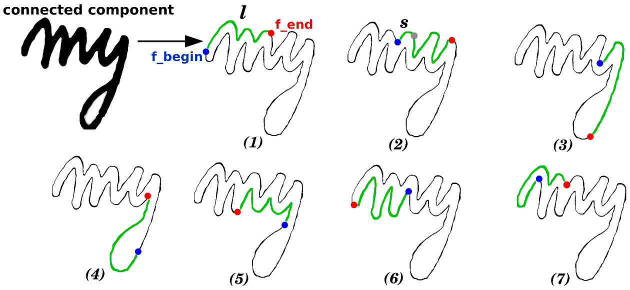

Figure 3. Fragments extraction on a connected component: l is the fragment length and s is the size of the overlap

Codebook generation

The codebook generation step aims at finding a collection of similar contour fragments that are most typical of each class. In the proposed system, this stage is performed using a 2D Self-Organizing Map (SOM) [2]. This clustering step enables to generate a codebook gathering the most representative fragments of each class. The definition of the classes present in the codebook depends on the application (writing type or script identification).

Classification process

As mentioned before, the classification of a text block is based on the classification of its connected components. An overview of the approach is presented in Figure 4.

For each connected component of the block, fragments are extracted and for each fragment of the connected component, the nearest fragment in the codebook is identified using an euclidean distance. For this computation, each fragment is described by its Chain Code Histogram (CCH) which is a eight dimensional histogram which shows the probability of each direction. Hence, the feature vector is a eight dimensional histogram which shows the probability of each direction. The number of occurrences of each codebook fragment in the external contour of the component is computed. This leads to a normalized histogram of occurrences representing the feature vector for the classification. The connected component level decision is taken by a MLP classifier. After the classification of the connected components, a majority vote is carried out to get the text block decision.

Application to writing type identification

The separation of text areas into printed and handwritten areas is an important step in the automatic transcription of complex documents and brings useful information for the script and the language identification. Writing type identification in a multilingual context is further more complicated, especially when a printed writing is cursive (for example with the Arabic). In order to tackle the difficulty of discriminating printed and handwritten text in the presence of different scripts (in our case Latin and Arabic scripts), we generate a 15×15 codebook gathering fragments in the different kinds of text (the different scripts in both writing types). [13] has shown that the combination of classifiers can increase the robustness and the performance of the classification. As a consequence, we generate a set of codebooks with various configurations (sizes of fragments) in order to combine the decisions of different systems. The size of the codebook has been chosen by experimenting different configuration from 5x5 to 30x30 and the best results were obtained with 15x15.

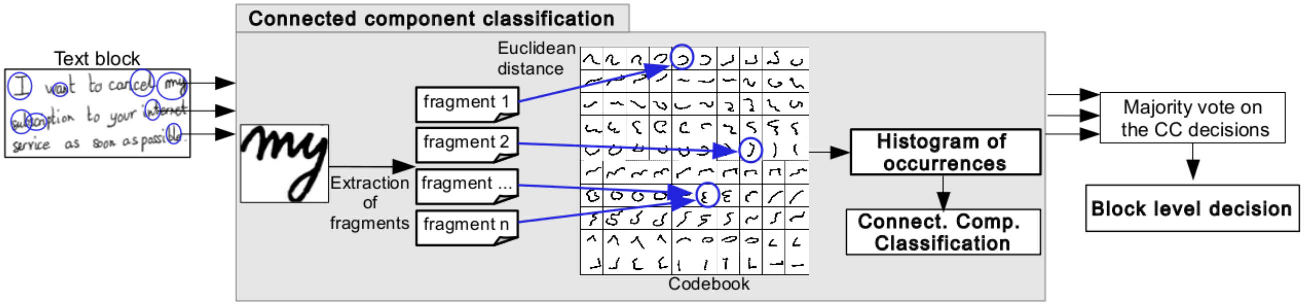
Connected component classification
Text block
I want to cancel my subscription to your internet service as soon as possible
Euclidean distance
fragment 1
fragment 2
fragment ...
fragment n
Extraction of fragments
Codebook
Histogram of occurrences
Connect. Comp. Classification
Majority vote on the CC decisions
Block level decision

Figure 4. *Classification process of a text block : classification of the connected components using codebook of contour fragments*

We have experimentally selected three codebooks generated with Latin printed, Arabic printed, Latin handwritten and Arabic handwritten fragments (extracted on a selection of the MAURDOR training database). The codebooks are built using three different sizes of fragments which have been experimentally optimized: $l = 15$, $l = 10$ and $l = 8$ pixels with an overlap of $s = 5$ pixels. Three MLPs (trained on the connected components of the MAURDOR training database) are combined to obtain the writing type decision at the connected component level. The sum combination rule is chosen to combine the three MLP outputs. Then, a majority vote is applied on the connected component level decisions to identify the writing type at the block level.

Once the writing type identified for each text block, we can proceed to the script identification taking advantage of the writing type information to adapt the approach.

Application to script identification

In languages of different scripts, characters are different, but ligatures between characters and words can also be discriminative. Consequently, the aim of this stage can be tackled in the same way as the writing type identification problem. Therefore, the system for printed/hand-written discrimination has been adapted to perform the script discrimination. The system takes into account the writing type information provided by the previous step in order to use expert codebooks and to specialize the decision process for each writing type. An overview of the proposed approach for script identification is presented in Figure 5.

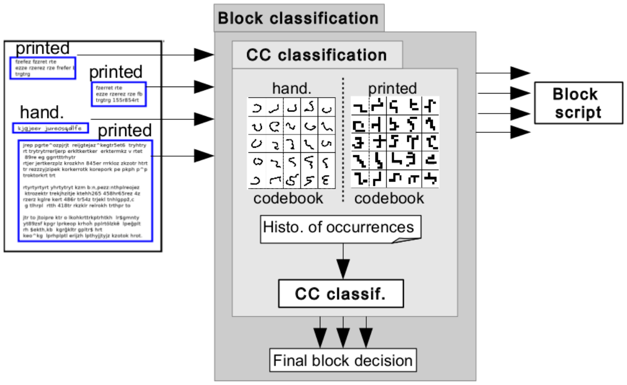

Figure 5. *Approach for script identification of a document: block classification based on expert codebooks of contour fragments*

The system uses the writing type information of the block to select the appropriate set of expert codebooks (codebooks specialized with printed or handwritten fragments) coupled with the corresponding MLPs. An expert codebook is a codebook gathering fragments of contours for one specific writing type (handwritten or printed). A set of expert codebooks is generated in the same way as writing type identification system, separating codebooks gathering handwritten fragments and codebooks gathering printed fragments (in both Latin and Arabic). Different configurations were tested and we chose empirically to use two sets of expert codebooks: one set with a size of fragments of $l = 10$ pixels and the other set with a size of fragments of $l = 30$ pixels.

Experimental results for writing type and script identification are fully detailed in section .

4. Language identification system

Languages sharing the same alphabet are difficult to discriminate using physical descriptors (such as English and French languages). In this latter example, the small specificities (i.e. presence or absence of accentuated characters) are not sufficient to reliably discriminate the shapes based on physical descriptors. Therefore, we have turned toward the use of textual descriptors as for language identification methods on electronic documents. One could use dictionary-based approach but this kind of approach requires a perfect recognition of the text in order to find the correct words in the dictionary. Another strategy is to perform a statistical analysis of characters distribution and sequences of characters distribution. Indeed, some characters are more frequently used depending of the language. For example, character '*W*' is used in a lot of common words in the English language, whereas there are less than 230 French words (that are not everyday words) containing this character. The same phenomenon can be observed for couples of characters. Moreover, the language analysis literature shows that *n*-gram analysis are efficient for digital document language identification.

Based on this observation, the proposed language identification system relies on the analysis of characters and n-gram (sequence of *n* characters) of an OCR output. We assume that the frequencies of some particular characters and some particular n-grams are strong characteristics of a language, even with errors in the transcription generated by the recognition engine. *n*-gram with $n > 2$ can be even more discriminative but need to ensure having correct sequences of *n* characters.

The key idea is to always use the same OCR for the extraction of n-grams distributions and during the recognition in order to replicate the same transcriptions errors. We use the LITIS

OCR based on HMM with variable state number, described in [8]. Since the language is unknown during recognition, this OCR is a language free version working at the character level (without any language model nor dictionary).

Overview of the approach

An overview of the proposed approach is presented in Figure 6.

First, a printed/handwritten Latin OCR is applied on the text blocks of the document in order to get separately the printed transcription and the handwritten transcription of the document. Language profiles are estimated on both transcriptions and for each language (French and English). The decision process relies on a comparison step of the different profiles measuring distances between the document profiles and profiles estimated on a training set.

Recognition engine

A document recognition engine is needed in order to estimate the language profiles. It is applied on each text block so that the transcription of the document is available. The recognition engine used to perform this task works on line images. We need to detect and segment the text lines contained in each text block. The line segmentation approach used to handle this problem is a modified version of the method detailed in [10]. The approach is based on an Adaptive Local Connectivity Map (ALCM) obtained applying a steerable directional filter on the image. Text line patterns in term of connected components are revealed using a local adaptive threshold on the ALCM. Text lines are extracted by collecting the connected components corresponding to a location mask.

Feature vectors are then computed on the text lines in order to feed the recognition engine. The features extracted from the line images are histograms of oriented gradients [11] computed in a sliding window applied along each text line. Feature vectors are given to a recognition engine based on Hidden Markov Models (HMM) of characters. For each text line we use the appropriate set of Latin models (typed or handwritten). The textual content of each line is decoded using Viterbi decoding without contextual resources as it is the case for standard recognizer (no dictionary, no language model used). A detailed description of the recognition engine is given in [8].

Language profile estimation

To select the appropriate language according to the n-gram distribution of characters, we need to estimate the language profiles (the distribution of characters and n-grams for each language). A language profile is estimated by recognizing the content of a document set of this language and estimate the character frequencies on the resulting transcription. Thanks to the previous printed/handwritten discrimination, we can refine the representation by defining two profiles for each language : a printed profile and a handwritten profile. In the Latin alphabet, we have to discriminate French from English. Hence, we get 4 profiles: French-hand, French-printed, English-hand and English-printed. These profiles are estimated on the documents from the MAURDOR training dataset (see section).

Decision process

The text content of a document is recognized using the same OCR as for language profile estimation. Then, the document profiles of characters and/or n-grams are generated for both handwritten and printed characters. Handwritten document profiles are compared with the set of hand-profiles (here, the French-hand and the English-hand) and the printed ones, with the set of printed profiles. The profile comparison is made by a weighted χ^2 like score to measure the distance between the document profile Pr_{doc} and the languages ones Pr_{lang}:

$$Score_{lang} = \sum_{b \in Pr_{doc}} \frac{\left(Pr_{doc}(b) - Pr_{lang}(b)\right)^2}{Pr_{lang}(b)} \times weight(b) \qquad (1)$$

The $weight(b)$ is the absolute difference between frequencies of character or n-gram b in the French and the English profiles, given by $weight(b) = |Pr_{eng}(b) - Pr_{fr}(b)|$. More generally, this is a coefficient that maximizes the contribution of most discriminative characters or n-grams. A character or a n-gram which is very frequent in a given language but rare in the other will have a strong influence in the computation of the score.

5. Experimental results

The system is evaluated on two sets of documents used during the MAURDOR campaigns [6]. These campaigns were led to evaluate the progress in automatic reading of heterogeneous documents and made an important step beyond other existing evaluation campaigns [1, 7] regarding the volume and the heterogeneity of the documents to be processed. Writing type and language identification constitute two sub-tasks that were evaluated during the MAURDOR campaigns. The results of our system are compared with the results of the participants of the second campaign that occurred in November 2013. In this section, the MAURDOR database is presented, the metrics are described, and the results are exposed.

The MAURDOR database

The MAURDOR database is composed of heterogeneous documents in their layout, their content or their quality. . . The kind of documents that can be encountered in the MAURDOR database are the following :

- Blank or filled in (by hand) forms;
- Printed business documents (invoice, bill, receipt, contract, legal or administrative document, etc.);
- Catalog pages, newspaper articles;
- Graphical documents (maps, drawings, posters, tables of digits, schemes,etc.);
- Private handwritten correspondences (invitation letter, post-it, etc.);
- Printed business correspondences.

Fonts and writings are different across documents and they are digitized using various digitizers at various resolutions (but mostly at 200 dpi). The documents are either in French, Arabic or English. Figure 7 contains some examples of documents and Figure 8 shows some examples of text regions. The corpus is composed of 6000 training documents and two sets of 1000 documents for the evaluations.

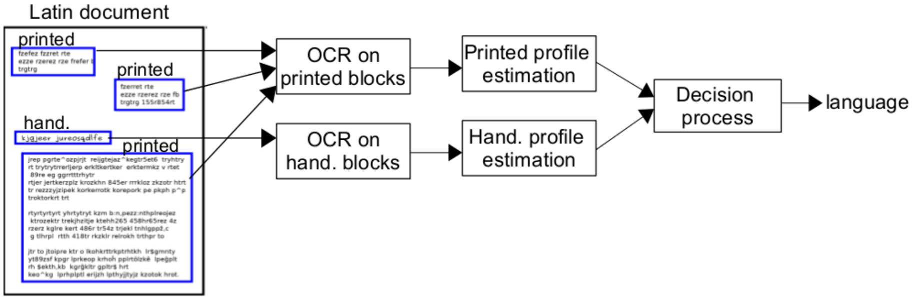

Figure 6. *Approach for language identification of a document: estimation of language profiles using the OCR transcription of the document*

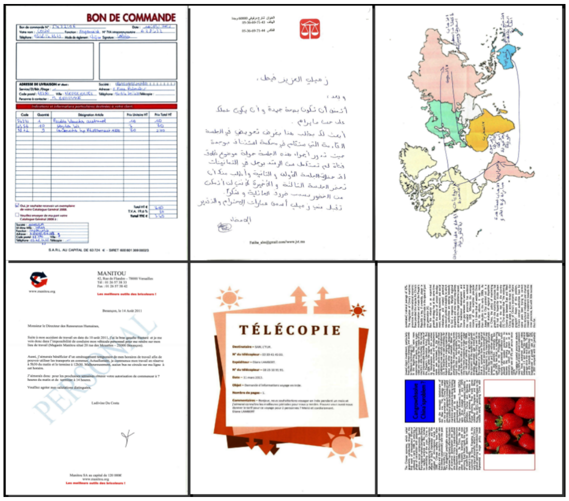

Figure 7. *Example of documents used in the MAURDOR campaigns*

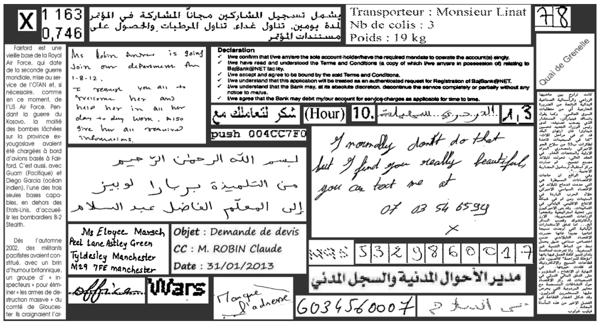
1 163
0,746
Transporteur : Monsieur Linat
Nb de colis : 3
Poids : 19 kg
(Hour)
push 004CC7F0
I normally don't do that but I find you really beautiful you can text me at 07 03 54 6599
Ms Elouge Mavsch
Peel Lane Astley Green
Tyldesley Manchester
M29 7FE manchester
Objet : Demande de devis
CC : M. ROBIN Claude
Date : 31/01/2013
Wars
603456007
Quai de Grenelle

Figure 8. *Example of text areas used in the MAURDOR campaigns*

The metrics

The evaluation of writing type and language identification were conducted using the classical Precision/Recall measure. A silence criterion has also been defined by the French National Metrology and Testing Laboratory (LNE) to evaluate the rejection ability of the methods. The Silence rate is the proportion of text areas that has been rejected by the algorithm. The system described in this paper is configured to always take a decision and does not generate any silence. We complete these metrics with the classical accuracy measure to present global performance and to compare with the state of the art.

Writing type and script identification experimental results

In this section, the results of the proposed systems for the writing type and script identification on the two evaluation datasets are presented. Each evaluation dataset is composed of 1000 documents.

Results of the writing type identification system

For the writing type identification task, inputs are documents with the position of all text blocks. Global results on the writing type identification as well as results per script are presented on Table 2.

Table 2: Writing type identification : Results of our system for the writing identification on the two campaigns

	Accuracy (%)		
	Global	Latin	Arabic
Campaign 1	92.00	91.30	94.21
Campaign 2	93.50	93.40	94.03

The system is quite stable between both campaigns and the results are encouraging regarding the heterogeneity of the corpus. Comparing with the state of the art, our approach achieves lower performance than approaches focused on one script (around 98% of accuracy [41, 39, 37]). However performance are difficult to compare when datasets are different, the difficulty of the issue being different from a dataset to another. Nevertheless, we can compare the results of our system with the results published in [40] evaluated on the documents of the first MAURDOR campaign. In this paper, two bonzaiboost systems were evaluated, the first system achieving 91.10% of accuracy and the second one reaching an accuracy of 94.07%. Comparatively, our system takes place between the two bonzaiboost systems with an accuracy of 92.00%.

We have also performed a statistical analysis of the errors produced by the system according to the number of characters in the blocks. First, let us analyze the block distribution in the two datasets. Figure 9 represents the distribution of blocks in the ground truth according to the number of characters. We can notice that approximately 70% of text blocks in the MAURDOR dataset have less than 20 characters ($\approx$ 40% of blocks having less than 10 characters). These statistics indicate that a majority of blocks contains few words and are more difficult to identify correctly.

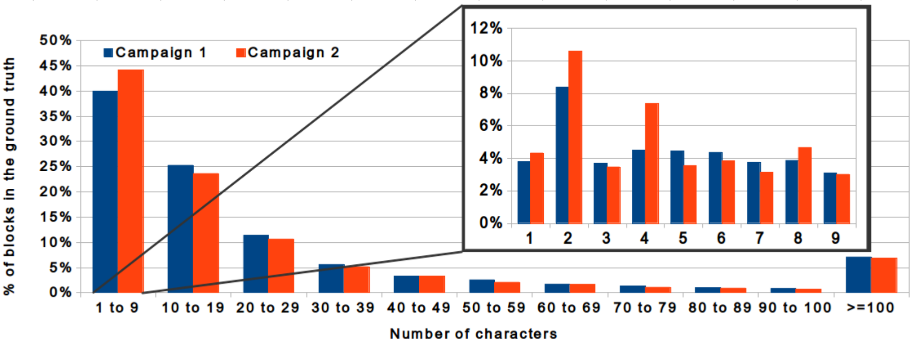

Figure 9. Distribution of blocks in the ground truth according to the number of characters for both campaigns datasets

When we look at Figure 10, representing the distribution of errors according to the number of characters in a block, we can see that the system makes more mistakes on blocks with less than 10 characters (12% and 8% of these blocks produced errors on the two campaigns). We can also notice that, as expected, the blocks with only one character generate most of the errors (23% and 16% of mistakes on these blocks on the two campaigns).

Finally, we can compare the results of our system with those of the participants of the second MAURDOR campaign in November 2013. Table 3 represents the global results of the task of writing type identification. The systems have first been designed in the MAURDOR context. In this paper, we present upgraded versions of the systems submitted for the second MAURDOR campaign. To see the improvement, we compare the current performance of our systems with the official campaign results. The systems called ”LITIS_1” and ”LITIS_2” are the systems used by LITIS during the campaign. These two systems are based on codebooks and include silence. ”Participant_1” denotes the other participant of the MAURDOR campaign. The system called ”This_work” refers to the system presented in this paper. One can see in Table 3 that the system LITIS_1 obtains the best precision but rejects more often, reducing its recall performance. If we look at our last system This_work, we can notice that without reject our system still achieves better performance than the other campaign participants.

Table 3: Writing type identification : Comparison with other participants on the second MAURDOR campaign (global results)

System	P (%)	R (%)	Sil (%)
LITIS_1	**96.11**	85.43	11.12
LITIS_2	95.55	86.39	9.58
Participant_1	93.30	93.16	0.15
This_work	93.50	**93.50**	0.00

Results of the script identification system

For the script identification task, inputs are documents with the position of all text blocks and the associated writing type. The system described in this paper for the script identification is evaluated on the two sets of documents of the MAURDOR campaigns. There is no possible comparison with other participants since this task was not evaluated during the campaigns. Global results on the script identification as well as results per writing type are presented on Table 4. We can see that global results are quite stable over the two campaigns.

Regarding the state of the art, the performance of our approach are slightly lower than the approaches working on both printed and handwritten documents (accuracy around 95%). However it seems that datasets used to evaluate state of the art approaches do not exhibit as much variability as the MAURDOR dataset (e.g. postal images, IAM-DB dataset).

Table 4: Script identification : Results on the two campaigns

	Accuracy (%)		
	Global	**Printed**	**Hand.**
Campaign 1	93.84	93.47	95.72
Campaign 2	92.51	91.92	94.93

This system is used in the evaluation of the language identification task. Therefore, more detailed results are presented in the following subsection.

Language identification experimental results

Like for script identification, inputs are documents with the position of all text blocks and the associated writing type. The important amount of small blocks in the dataset (70% of text blocks have less than 20 characters) led us to adapt our strategy by estimating bi-gram or character distributions at the document level in order to have a sufficient amount of information. We evaluate two main configurations on the two campaign datasets:

- **Code + distrib :** The system described above, made of script identification (Arabic/Latin) using codebook and language identification (French/English) using distributions of Latin OCR output
- **Full distrib :** Script identification (Arabic/Latin) and language identification (French/English) are both performed using the distributions of Latin OCR output

For this last configuration, as for the other ones, only a Latin OCR is used, even on Arabic documents. The discrimination between Arabic and Latin documents relies on the errors of the Latin OCR on these documents.

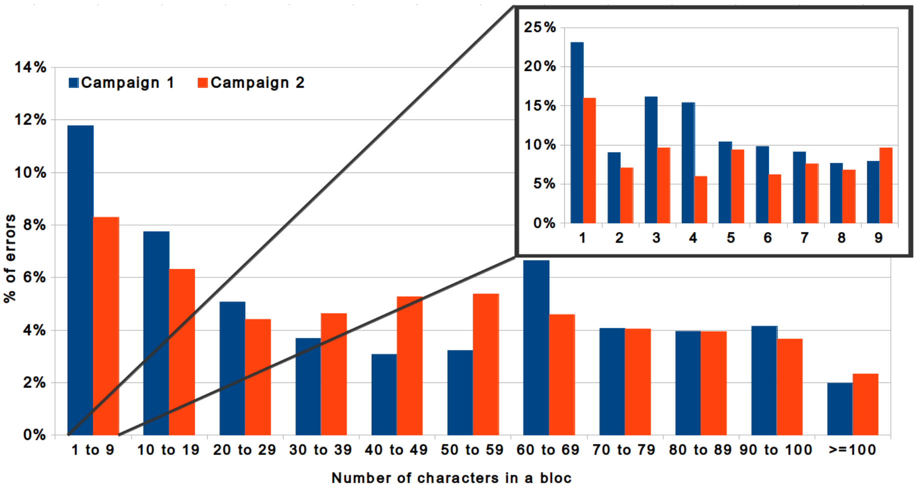

Figure 10. *Distribution of errors according to the number of characters in a block*

All the distribution based stages have been tested with several configurations. We introduce some notations to characterize the system configuration:

- **CHAR :** the system uses the character distributions
- **2G :** the system uses the bi-gram of character distributions
- **3G :** the system uses the 3-gram of character distributions
- **CHAR+2G :** character profiles and bi-gram profiles are both used to compute distances to a language profile (the distance is the sum of the character distance and the bi-gram distance)

Evaluation of the system configurations

Table 5 reports the language identification performance of the full distribution system using characters, bi-gram or 3-gram of characters. Character profiles might be more robust on difficult documents than bi-gram ones because of the OCR output reliability. Indeed, it is more difficult to get stability in the accuracy of having two consecutive characters. However, bi-gram profiles encode more knowledge and might be better for good quality documents. To evaluate the limits of knowledge encoding, we test the system with 3-gram profiles.

Table 5: Language identification : evaluation of the full distribution system using characters, bi-gram or 3-gram of characters

System	Accuracy (%)	
	Campaign 1	**Campaign 2**
Full distrib CHAR	78.32	82.05
Full distrib 2G	**86.95**	**87.23**
Full distrib 3G	83.34	77.20
Full distrib CHAR+2G	83.64	84.66

We can see that using character profiles provides lower performance than character bi-grams. We think characters can be discriminant only for a small subset of them like the 'w' or the 'y' for the French/English discrimination. But for all other cases n-grams are obviously more discriminant. Moreover, bi-grams will also encode the discriminative power of the discriminative subset of characters. So the character profile does not bring information that is not already in the bi-gram profile. Reasoning this way may encourage us explore 3-gram, 4-gram and more. But on the other hand the analysis of OCR outputs (with errors) instead of a reliable text transcription is less likely to have stability in 3-gram or 4-gram. This assumption is globally confirmed by the results of the systems with 3-gram profiles and can explain the lower performance obtained compared to using bi-grams.

Character bi-grams configuration has been selected for the distribution based stages. The system using codebooks for script identification is compared with the full distribution system and performance are presented in Table 6.

Table 6: Language identification : System comparison on the documents of the first and the second MAURDOR campaign

System	Accuracy (%)		
	Global	**Printed**	**Hand.**
Campaign 1			
Code + distrib 2G	**88.41**	**87.83**	**90.16**
Full distrib 2G	86.95	87.00	86.78
Campaign 2			
Code + distrib 2G	**87.36**	86.28	**90.63**
Full distrib 2G	87.23	**87.03**	87.82

We can notice that adding codebook information for the script discrimination increases the performance by a small amount on the two campaigns. The drop of performance with the full distribution system is due to Arabic documents. The discrimination between Arabic and Latin relies on the errors of the Latin OCR (and only errors for Arabic). In this case, stability in errors in order to get stable bi-grams is difficult. As a consequence, the system selected for language identification is the bi-gram version combined with the codebook approach for the script identification part. Comparing with the state of the art, our approach achieves better performance on handwritten text than [25] which reached an accuracy rate of 85% on handwritten documents. Performance on printed text are slightly below the state of the art, but global results are encouraging regarding the complexity of the problem and the fact that this is the first time that language identification on heterogeneous documents is performed.

Because we use distributions of characters, one can wonder what is the minimal number of characters (or bi-gram) in a document in order to get correct language identification. This is what we try to evaluate in this paragraph measuring the percentage of miss-classified documents according to the number of characters. As depicted on Figure 11, 60% of documents are globally equally distributed, from 0 to 1000 characters per document. Figure 12 represents the percentage of documents where 90% to 100% of text blocks are miss-classified. Knowing that the average miss-classification rate on the whole dataset is between 10% and 15%, we can conclude that blocks that contain less than 400 characters are more likely to be miss-classified. But we can not identify a real critical number of character per document that ensure a miss-classification.

Comparison with the Google plug-in results

We evaluate the performance of the Google plug-in on the ground-truth transcriptions of the two MAURDOR datasets in order to estimate the complexity of the dataset. The plug-in is first evaluated at the block level on French and English transcriptions. We compare the performance of the plug-in with the performance of our system configured to take decisions at the block level. Results are presented in Table 7. As expected, the performance of our system drop dramatically since the MAURDOR dataset contains a lot of tiny blocks of text and the language identification on this kind of data is much more complicated than having a full page of text content. On the other hand, even with the ground-truth transcription, the Google plug-in does not seem able to perform language identification at the block level. The plug-in fails to extract features on small blocks, however these blocks represent the majority of the MAURDOR dataset. The Google plug-in achieves lower performance than our approach that does not have access to the transcription and performs the recognition.

Table 7: Google plug-in results : Results at the block level on the ground-truth transcriptions of the two campaigns

	Accuracy (%)	
System	**Campaign 1**	**Campaign 2**
Google plug-in	42.41	39.91
Codebook + distrib 2G	**73.05**	**70.54**

The plug-in failing to detect the language at the block level, we evaluate the performance at the document level by concatenating the ground-truth transcription of each blocks. Results are given in Table 8. As predicted, performance improve considerably at the page level. Nevertheless, we could expect better accuracy knowing that the system is evaluated on the ground-truth transcriptions and not on the OCR outputs.

These results allow us to conclude that language identification on the MAURDOR dataset is a complicated issue. We succeed to obtain performance close to the Google plug-in while our system does not have access to the ground-truth transcriptions. This shows the effectiveness of our approach for the language identification on a complex dataset.

Table 8: Google plug-in results : Results at the page level on the ground-truth transcriptions of the two campaigns

	Accuracy (%)	
Global	**Campaign 1**	**Campaign 2**
Google plug-in	86.22	**88.32**
Codebook + distrib 2G	**88.41**	87.36

Comparison with the MAURDOR campaign results

We compare the current performance of our systems with the official campaign results. The Tables 9 and 10 present the global and per language performance respectively of our system submitted to the competition and the best configurations of our systems at this time. LITIS 1 and LITIS 2 are respectively the "code + distrib" and the "full distrib" versions submitted for the evaluation campaign. The systems LITIS 1 and LITIS 2 outperform the other campaign participant. Our system LITIS 2 was ranked first for this competition. However, we can see that the evolutions made after this campaign increase significantly the results. The use of codebook for the script identification is now slightly better than the analysis performed by exploiting the Latin OCR.

Table 9: Language identification : Results on the documents of the second MAURDOR campaign

System	**P (%)**	**R (%)**	**Sil (%)**
LITIS 1	78.95	71.99	8.97
LITIS 2	83.65	83.65	0.00
Participant_1	57.88	55.66	4.00
code + distrib BGχ^2 W	**87.36**	**87.36**	0.00
full distrib χ^2 W	87.23	87.23	0.00

Results of the entire system for language identification

In this section we evaluate the entire system including writing type, script and language identification. We measure here the capacity of the system to detect the correct language having only the block localization (without the writing type nor the script information). Therefore, the difference with respect to the previous results is that script and language identification does not benefit from the ground truth writing type, but only from the output of our writing type method. The system evaluated is the system using the codebook approach to perform the script identification. The results are given in Table 11 and show the robustness of the system. We can notice a loss ranging between 0.66 and 1.36 points and we can see that the performance are close to the results obtained using the ground-truth information (Table 6). These results show that our system can efficiently identify the language of a document as well as the writing type of the different text regions in order to apply the correct OCR on the document thereafter. There is no

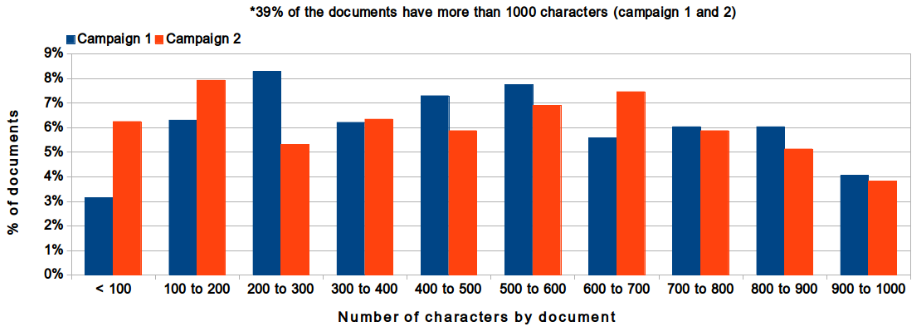

Figure 11. *Distribution of documents in the ground truth according to the number of characters*

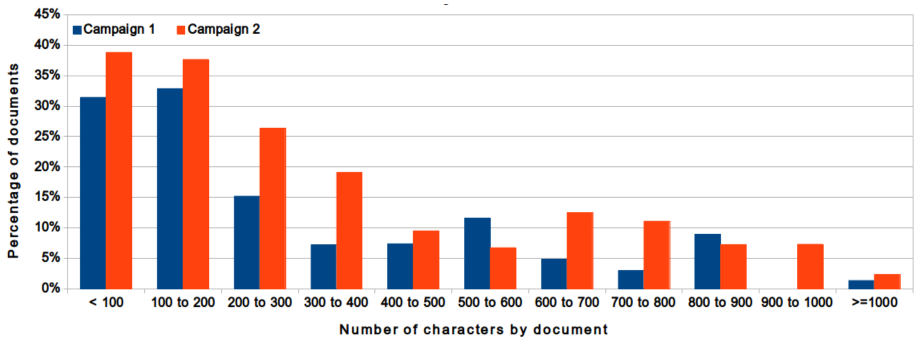

Figure 12. *Percentage of miss-classified documents according to the the number of characters (for documents having 90% to 100% of errors)*

possible comparison with the state of the art since, to the best of our knowledge, none of the literature approaches handle language identification as well as script and writing type identification.

Table 10: Language identification:Results on the documents of the second MAURDOR campaign per language

System	**P (%)**	**R (%)**	**S (%)**
Arabic			
LITIS 1	58.42	**96.03**	2.34
LITIS 2	75.64	86.92	0.00
Part_1	29.24	4.96	3.42
code+distrib	**80.45**	81.34	0.00
full distrib	70.70	91.80	0.00
English			
LITIS 1	**91.18**	56.17	10.89
LITIS 2	85.04	58.47	0.00
Part_1	25.00	0.05	4.53
code+distrib	87.10	**79.73**	0.00
full distrib	89.97	75.36	0.00
French			
LITIS 1	88.97	70.17	10.20
LITIS 2	86.10	92.37	0.00
Part_1	58.90	**93.16**	4.00
code+distrib	89.65	92.47	0.00
full distrib	**94.24**	90.50	0.00

Table 11: Writing type + Language identification : Results on the documents of the two campaigns

	Accuracy (%)		
	Global	**Printed**	**Hand.**
Campaign 1	87.05	86.71	88.05
Campaign 2	86.70	85.88	89.18

6. Discussion and future work

In this paper we have presented three complementary approaches devoted to writing type, script and language identification in complex mixed printed and handwritten documents. Writing type and script are identified thanks to a set of physical codebooks classified by a MLP. Language identification relies on an original statistical analysis of bi-grams of an OCR output. The results obtained on the MAURDOR dataset for the sub-tasks of writing type and language identification (including the script identification) compare favorably our systems to the other participants. The writing type identification is 93.50% accurate on the second campaign and the best language identification system relies on character bi-grams analysis (with the script identification made by the codebook approach) and achieves a precision rate of 87.36% on the same dataset.

Although efficient, our writing type identification system can be improved adding a preprocessing step in order to correct the inverse video, to remove the rule lines and improve the quality of the contour fragments. In the language identification system, we use an OCR at character level, that is the hardest way for text transcription. An alternative approach could be to use an OCR with both French and English language models and compare recognition scores to choose the correct language. Finally, our systems need to be evaluated on datasets with more scripts and more languages.

References

[1] E. Augustin, M. Carr, E. Grosicki, J. M. Brodin, E. Geoffrois and F. Preteux, *RIMES evaluation campaign for handwritten mail processing*, In Proceedings of IWFHR, 2006, 231–235

[2] M. Bulacu and L. Schomaker, *A comparison of clustering methods for writer identification and verification*, In Proceedings of the Eighth ICDAR,2005, 1275–1279

[3] L. Schomaker, K Franke and M Bulacu, *Using codebooks of fragmented connected-component contours in forensic and historic writer identification*, PRL, Volume 28, 2007, 719–727

[4] G. Ghiasi and R.W. Daly, *An efficient method for offline text independent writer identification*, ICPR,IEEE,2010,1245–1248

[5] J. Iivarinen and A. Visa, *Shape Recognition of Irregular Objects*, Intelligent Robots and Computer Vision XV: Algorithms, Techniques, Active Vision, and Materials Handling, Proc. SPIE 2904,1996,25–32

[6] MAURDOR campaign website, *http://www.maurdor-campaign.org/*

[7] B. Gatos, N. Stamatopoulos, G. Louloudis, *ICDAR 2009 Handwriting Segmentation Contest*, ICDAR, 2009, 1393–1397

[8] K. Ait Mohand, T. Paquet and N. Ragot, *Combining structure and parameter adaptation of HMMs for printed text recognition*, IEEE PAMI, 2014, 1716–1732

[9] P.Barlas, S. Adam, C. Chatelain, and T. Paquet, *A typed and handwritten text block segmentation system for heterogeneous and complex documents*, In DAS, 2014, 46–50

[10] Z. Shi, S. Setlur, V. Govindaraju,*A steerable directional local profile technique for extraction of handwritten Arabic text lines*, In 10th ICDAR, 2009, 176–180

[11] J. Rodriguez, F. Perronnin, *Local gradient histogram features for word spotting in unconstrained handwritten documents*, In ICFHR, 2008, 7-12

[12] D. Hebert, P. Barlas, C. Chatelain, S. Adam and T. Paquet, *Writing Type and Language Identification in Heterogeneous and Complex Documents*, ICFHR, 2014

[13] A. J. C. Sharkey, N. E. Sharkey, *How to improve the reliability of artificial neural networks*, In Technical Report CS-95-11, Department of Computer Science, University of Sheffield, 1995

[14] Google plug-in website for language detection, *http://code.google.com/p/language-detection/*

[15] L. Grothe, E. W. De Luca, A. Nürnberger, *A Comparative Study on Language Identification Methods*, In LREC, 2008

[16] B. Martins, M. J. Silva, *Language identification in web pages*, In Proceedings of the 2005 ACM symposium on Applied computing, 2005, 764–768

[17] E. Tromp, M. Pechenizkiy,*Graph-based n-gram language identification on short texts*, In Proc. 20th Machine Learning conference of Belgium and The Netherlands, May 2011, 27–34

[18] T. Dunning, *Statistical Identification of Language*, Techreport, 1994

[19] G. Grefenstette, *Comparing two language identification schemes*, JADT, 1995

[20] R. Řehůřek, M. Kolkus, *Language identification on the web: extending the dictionary method*, CICLing, 2009, 357–368

[21] W. B. Cavnar, J. M. Trenkle, *N-Gram-Based Text Categorization*, SDAIR, 1994, 161–175

[22] P. Sibun, A. L. Spitz, *Language determination: Natural language processing from scanned document images*, In ANLP, October 1994, 15–21

[23] L. Shijian, C. L. Tan, *Script and language identification in noisy and degraded document images*, IEEE PAMI, 2008, 14–24

[24] D. S. Lee, C. R. Nohl, H. S. Baird, *Language identification in complex, unoriented, and degraded document images*, In Series in Machine Perception and Artificial Intelligence, 1998, 17–39

[25] J. Hochberg, K. Bowers, M. Cannon, P. Kelly, *Script and Language Identification for Handwritten Document Images*, IJDAR, 1999, 45–52

[26] B. Waked, S. Bergler, C. Y. Suen, S. Khoury, *Skew detection, page segmentation, and script classification of printed document images*, IEEE SMC, October 1998, 4470–4475

[27] B. V. Dhandra, P. Nagabhushan, M. Hangarge, R. Hegadi, V. S. Malemath, *Script identification based on morphological reconstruction in document images*, PR, August 2006, 950–953

[28] L. Zhou , Y. Lu, C. Tan , *Bangla/English Script Identification based on Analysis of Connected Components Profiles*, In DAS, 2006, 243–254

[29] W. M. Pan, C. Y. Suen, T. D. Bui, *Script identification using steerable Gabor filters*, ICDAR, August 2005, 883–887

[30] A. M. Elgammal, M. A. Ismail, *Techniques for Language Identification for Hybrid Arabic-English Document Images*, ICDAR, 2001, 1100–1104

[31] I. Moalla, A. Elbaati, A. M. Alimi, A. M. Benhamadou, *Extraction of Arabic text from multilingual documents*, IEEE SMC, October 2002, vol. 4

[32] H. Ma, D. Doermann, *Gabor filter based multi-class classifier for scanned document images*, ICDAR, 2003, 968–968

[33] G. Zhu,X. Yu, Y. Li, D. Doermann, *Language identification for handwritten document images using a shape codebook*, PR, 2009, 3184–3191

[34] S. Kanoun, I. Moalla, A. Ennaji, A. M. Alimi, *Script Identification for Arabic and Latin Printed and handwritten Documents*, DAS, 2000, 159–165

[35] A. K. Echi, A. Sadani, A. Belad, *How to separate between Machine-Printed/Handwritten and Arabic/Latin Words?*, ELCVIA, 2014, 1-16

[36] J. Kumar, R. Prasad, H. Cao, W. Abd-Almageed, D. Doermann, P. Natarajan, *Shape codebook based handwritten and machine printed text zone extraction*, IS&T/SPIE Electronic Imaging, 2011, 787406–787406

[37] E. Kavallieratou, S. Stamatatos. *Discrimination of Machine-Printed from Handwritten Text Using Simple Structural Characteristics*, ICPR, 2004, 437–440

[38] K.C. Fan, L. S. Wang, Y. T. Tu, *Classification of machine-printed and handwritten texts using character block layout variance*, PR, 1998, 1275–1284

[39] Y. Zheng, H. Li, D. Doermann. *Machine Printed Text and Handwriting Identification in Noisy Document Images*, IEEE PAMI, March 2004, 337–353

[40] Y. Ricquebourg, C. Raymond, B. Poirriez, A. Lemaitre, B. Coasnon, *Boosting bonsai trees for handwritten/printed text discrimination*, DRR, 2014

[41] U. Patil, M. Begum. *Word Level Handwritten and Printed Text Separation Based on Shape Features*, IJETAE, April 2012

[42] J. K. Guo, M. Y. Ma, *Separating handwritten material from machine printed text using hidden Markov models*, ICDAR, 2001, 439–443

[43] K. Kuhnke, L. Simoncini, Z. M. Kovacs-V, *A system for machine-written and hand-written character distinction*, ICDAR, 1995, 811–814

[44] J. Koyama, A. Hirose, M. Kato, *Local-spectrum-based distinction between handwritten and machine-printed characters*, In Image Processing, October 2008, 1021–1024