2014 22nd International Conference on Pattern Recognition

Multipage Administrative Document Stream Segmentation

Hani Daher[a], Mohamed-Rafik Bouguelia[a], Abdel Belaïd[a] and Vincent Poulain D'Andecy[b*]
[a]Université de Lorraine - LORIA, UMR 7503
Vendoeuvre-Lès-Nancy, F-54506, France
[b]ITESOFT Groupe, 30470, Aimargues, France
Email: [a]{hani.daher, mohamed.bouguelia, abdel.belaid}@loria.fr
[b]Vincent.PoulaindAndecy@itesoft.com

***Abstract*—We propose in this paper a framework for the segmentation and classification of document streams. The framework is composed of two modules: segmentation and verification. The two modules use an incremental classifier which learns progressively along the stream. In the segmentation module a relationship between two consecutive pages is classified as either: continuity or rupture. Rupture is synonymous of a clear break, thus probably a complete document. If the classifier is uncertain on whether the relationship should be a continuity or a rupture, an over-segmentation is proposed and we consider that we have a fragment *i.e.portion* of a document. Both fragments and documents are sent to the verification module where additionally to the incremental classifier it includes a correction module. The classifier predicts the classes of fragments and documents. The predicted class represents a context which is used as a query to search for similar contexts in the correction module and correct the segmentation and verification results. Corrections are sent back to the segmentation and verification modules to learn the correct classes. Results on real world databases show the effectiveness and stability of our approach.**

I. Introduction

Various types of documents stream into organisations every day, from claims, forms, invoices, contracts etc. A never ending stream of papers, e-mails and faxes must be processed quickly, accurately and dispatched correctly to the appropriate recipients. Handling this stream of information manually by sorting documents and keying in data is a time consuming, costly and error-prone approach. One solution is to introduce page separators or machine readable marks like bar codes to indicate the end of a document. In the case of page separators this approach is also costly and intensive because they must be inserted before the scanning of pages and, if they are not to be reused, removed afterwards. In high volume operations, these costs can be staggering. In the case of bar codes they offer more accurate document identification, but also at high cost. The use of papers, ink and codes is not an easy task and it is costly also. Inserting these bar codes between the documents is error prone. Bar codes must be inserted correctly to ensure that the correct separator sheet is used. To solve this problem we propose an approach capable of segmenting streams of pages into documents without the need of information on the number of pages, the beginning and ending of a document. This paper is structured as follows: In section I-A we present a literature review on previous works related to the segmentation, classification and retrieval of documents. A description of the incremental classifier is presented in section II. A general view of the framework is illustrated in section III. The stream of pages and the complexity of the documents on which we are working are described in section IV. The segmentation module is described in details in section V. In section VI we briefly explain the page combination and classification module. In section VII we illustrate and analyse the results. In section VIII we end by the conclusion and perspectives.

A. Related work

To our knowledge there exists three categories of approaches used in document flow processing :

- Segmentation: corresponds to the partitioning of a stream into several subsets of documents
- Retrieval: corresponds to the matching of a query document against a set of documents in a database
- Classification: corresponds to the assignment of a document to a specific class or group

1) Segmentation: Thompson and Nickolov [1] use a bottom-up clustering algorithm to treat the problem of documents stream segmentation. Every page in the stream forms its own cluster, then increasingly pairs of clusters are grouped together by using a single-linkage criterion. The authors consider that pages, in a same document contain, a lot of similarities. In real world applications the content of pages may bear very little similarities. Meilender and Belaïd [2] propose a method similar to the variable horizon models (VHM) or multigrams used in speech recognition. It consists in maximizing the flow likelihood knowing all the Markov Models of the constituent elements *(i.e. pages)*. Since the calculation of the likelihood of all the pages flow is NP-complete, the solution is to use windows to reduce the number of observations. The proposed algorithm seeks to identify the document in order to isolate the beginning and the end from it and then place the segmentation points between the various identified documents. The first results obtained on a homogeneous document streams produced more than 75% precision and 90% recall. Schmidtler and Amtrup [3] use a Markov chain model. Single pages are characterized by bag of words. According to the authors, the discriminating features are located in the first and last page of a document. Therefore they model the document types by using three symbols: start, middle and end. multiclass SVMs are used and their scores are mapped into probabilities. The probable best sequence of documents is extracted by using an algorithm

1051-4651/14 $31.00 © 2014 IEEE
DOI 10.1109/ICPR.2014.176

similar to the beam search algorithm in [12]. Gordo et al.[4] propose a probabilistic model to treat the problem of multipage document segmentation and classification. The probabilistic model is then incorporated into a page stream segmentation algorithm. The segmentation algorithm is based on dynamic programming and is used to retrieve the segmentation with the best score.

2) Retrieval: Rusiñol et al. [5] treat the problem of multipage documents retrieval by proposing two fusion strategies, namely the early and late fusion to combine sets of pages into one document. The first method represents a page by a histogram reflecting the occurrence of words in all the pages. The second method calculates the similarity between the current page and pages in a database. Pages with the best scores are combined to form a document. Kumar and Doermann [6] present a method based on bag of words. It relies heavily on the structure of the document and is applied to solve a single-page document retrieval problem. Shin and Doermann [7] segment pages into blocks that are characterized by conceptual and geometric features. The distance between the query image and other images in the database is achieved by mapping the blocks of images. The drawback of this method is that the extracted features are very specific to a category of documents.

3) Classification: Gordo and Perronnin [9] treat the problem of multipage documents classification. Every document is composed of different classes of pages including: papers, insurances, invoices etc. By using the bag of words principle, they propose a bag of pages approach where a document is represented by a histogram that includes the number of occurrences of these classes. Shin et al. [8] use the layout and structure of documents as features to apply the classification. Features such as column structures, percentage of text and non-text are extracted. This method does not propose a strategy for multipage documents classification. It is well adapted for single page documents. Most of the previous methods use static classifiers, they make the assumption that a training dataset is always available, however this is not the case in real world applications where companies have to deal with continuous streams of documents. To solve this problem we propose to use an incremental classifier to accommodate new classes of documents without the need for a fixed training dataset [10]. The fusion method proposed in [5][4] will be used in our algorithm to fuse pages into documents.

II. Overview of the Incremental classifier

In this section, we describe the incremental classifier [11] used in the segmentation and verification modules. The classification model is presented as a set of labelled data-representatives Y. Each data-representative $y_k \in Y$ is a feature-vector which is continuously updated by the classifier. When a data x is given as an input to the classifier, it either predicts the class c of x or rejects it. Let $P(c|x)$ the probability that an element x belongs to a class c. It is determined as:

$$P(c|x) = \frac{\sum_{(y_k,c_{y_k})\in KNN(x)} f(y_k, c_{y_k})}{K} \quad (1)$$

where $f(y_k, c_{y_k}) = \begin{cases} 1, & if\ c_{y_k} = c \\ 0, & otherwise \end{cases}$

$KNN(x)$ is the set of the K representatives closer to x. Let $c_1 = \underset{c}{\operatorname{argmax}} P(c|x)$ and $c_2 = \underset{c\neq c_1}{\operatorname{argmax}} P(c|x)$, c_1 and c_2 are respectively the first and second most probable classes, given an element x, such that $P(c_1|x) \geq P(c_2|x)$. The predicted class of x is c_1. Let the quantity $Q_x = P(c_1|x) - P(c_2|x)$. A small Q_x value indicates that the classifier is uncertain about whether the class of x is c_1 or c_2. In this case, x should be rejected. To decide if an element x should be rejected, we define a small confidence value δ such that if $Q_x < \delta$ then x is rejected. Otherwise, x is classified as c_1.

Given a classified data x. The representatives set Y is updated by modifying the closest data-representatives $y_k \in Y$ to x, to be either closer or farther from x, depending on whether they are labelled similarly of differently from x. If x is far enough from all the existing data-representatives (given a distance threshold), then x becomes a new representative (it is added to Y). For more details about rejecting uncertain data and updating representatives, refer to [11].

III. Proposed approach

The general framework is composed of two modules; they are independent, but their results are combined to make a decision (Fig.1).

Segmentation module: Generic features designed for administrative documents are first extracted, they are not intended for document classification instead they are specifically used by an incremental classifier to classify the relation between consecutive pages as either continuity or rupture. In case of a continuity, pages are combined together and stream analysis continues (Fig.1a). In case of a rupture, a document d is formed by the combined pages and sent to the verification module. The uncertainty measure Q_x is used to verify if there is a confusion or whether a relation should be classified as continuity or rupture. In the case of an uncertainty, a rejection r_e is issued and a fragment or a list of fragments f^* is formed and sent to the verification module (Fig.1b).

Verification module: Documents d and fragments f^* are characterized by Bag-Of-Words where the frequency of each word is used as a descriptor to classify the documents. The verification module uses another instance of the same incremental classifier which role is to predict a context i.e. a class label to f and d with a confidence score (probability) of belonging to the class. The context will play the role of a query to correct the segmentation and verification mistakes in the correction module (Fig.1c). The uncertainty measure Q_x is not needed in the verification module since the documents or fragments are always sent to the correction module where the output is considered to be always right. As output we might have *one* to *many* documents d^* (Fig.1d). At the moment the correction module is simulated by a domain expert to provide the correct segmentation and verification labels based on the algorithm illustrated in [11]. We will show in the next sections how the incremental classifier is used in the segmentation and verification modules.

IV. Stream of pages

The stream on which the analysis is done is composed of heterogeneous documents (Invoices, Insurances etc.) Fig.2

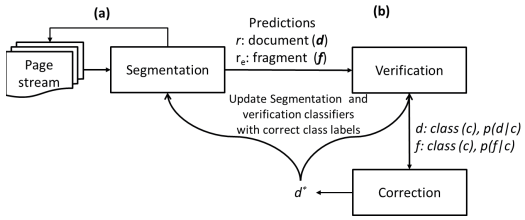


Fig. 1: Segmentation and verification modules

illustrates an example of four classes of documents. The arrival order of multipage documents in the stream is random. But In a same document, pages are always ordered, which allows us to extract elements of continuity such as page numbers, invoice numbers, dates, identification numbers, etc. in order to determine whether two successive pages in the stream belong to the same document.

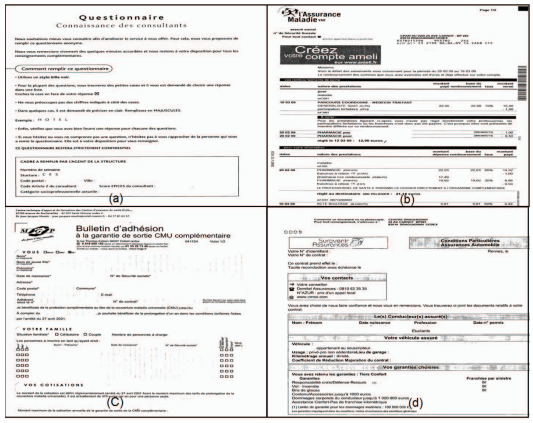


Fig. 2: (a) Questionnaire, (b) Payment form, (c) Newsletter, (d) Insurance contract

A. Structure variability of pages inside a same document

Fig.3 illustrates two pages of a same document. The structure of the pages is variable. It is difficult to decide if these pages belong to the same document based on visual features only. It is from the reference number, which is a textual descriptor, that we were able to infer that these two pages belong to the same document.

V. Segmentation

A detailed explanation of the segmentation module is explained in the following sections.

A. Preprocessing

Images are OCR-ed, stop words are removed and every page is presented by an *XML* format. Every block, line and word, are identified by their bounding box coordinates, named *(top, left), (bottom, right)*.

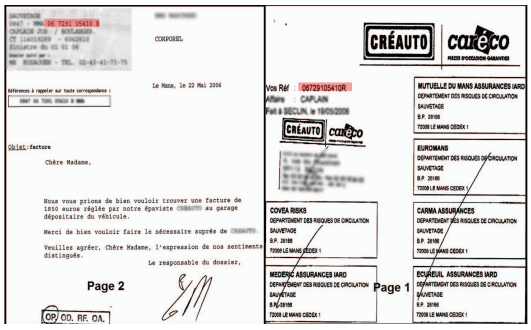


Fig. 3: Structure variability inside a same document

B. Feature extraction

Let $P = p_1, ..., p_n$ be a stream of pages. We illustrate in this section the steps we followed to characterize every page p_i and to model the relationships between a pair of pages p_i and p_{i+1} .

1) Descriptor classes: The analysis of the document stream revealed several types of descriptors such as fax, dates, codes, numbers, Ids and pages (see Table. I). Each column indicates the different instantiations of the type of descriptor in the documents. These descriptors, when repeated on two successive pages, can inform on a potential continuity or rupture between the pages.

TABLE I: 6 Classes of textual descriptors

Fax	Date	Code	Number	Id	Page
Date	Expedition	Account	Number	Global Id	Page Number
Number	Assignation	Receiver	Folder	User Id	Page Font
Page number	Mission	Shipping	Social security	-	Margin
Hour	Deadline	Matriculation	Client	-	Item
Telephone	Invoice	Zip code	Order	-	-

2) Retained descriptors: As instances of descriptors are numerous, difficult to extract and can still expand with the arrival of new classes of documents in the stream, we extract the descriptor values regardless of their instances. This allows us to reduce the dimension of the descriptors to 9 entities (see Table. II). For example, the descriptor f_1 (date) represents all instances of dates that are found by regular expressions.

TABLE II: Retained descriptors

Feature	Description	Type
f_1	Date	set of alphanumeric strings
f_2	Hour	set of alphanumeric strings
f_3	Telephone number	set of numeric strings
f_4	Zip code	set of numeric strings of length = 5
f_5	Alphanumeric	set of alphanumeric strings
f_6	Numeric	set of integer strings of length > 6
f_7	Salutation	set of strings
f_8	Page number	Single value,numeric
f_9	Margin	Single value,numeric

3) Descriptors extraction: Every descriptor among $f_1, ..., f_7$ is extracted with one or more regular expressions that reflect its formats. Thus, it is limited only to its value without consideration of its meaning given by a surrounding keyword. The page number, because of shortness

and confusion with other numbers, is first identified by a keyword such as P., Page, Pg, etc. (called context) in order to locate efficiently the number. Fig.4 illustrates an example of descriptors extraction by regular expressions: f_1 is composed of a set of 4 dates $f_1 = \{d_1, d_2, d_3, d_4\}$. The regular expression didn't find any descriptor of type f_2. For the page number f_8 in Fig.4 the regular expression found the context. The value 1 located on the right side of the context is attributed to f_8. f_9 represents the width of the largest block.

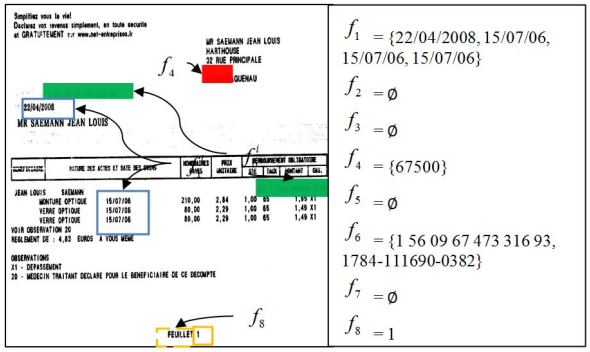
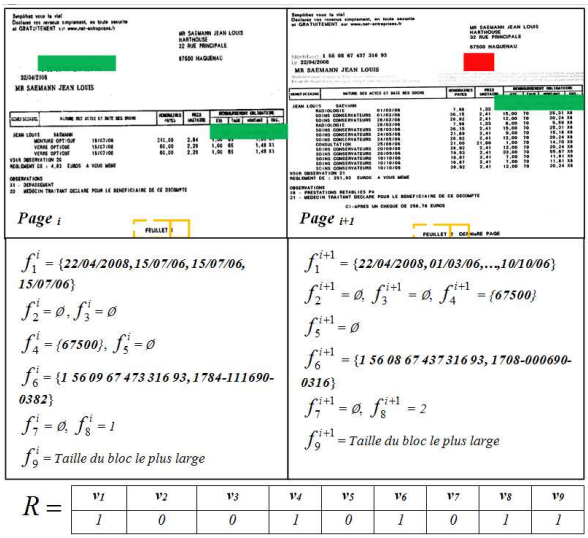


Fig. 4: Descriptor extraction by regular expressions

4) Relationship Modelling between two successive pages: After extracting the descriptors, we model the relationships between pairs of consecutive pages p_i and p_{i+1}. We noticed that a pairwise comparison of the descriptors between two pages with a majority of descriptors that points out to a continuity or rupture does not necessarily lead to a correct continuity or rupture decision. It is for this reason that we have created a representative vector whose components are the results of pairwise comparison of descriptors. The representative vector is used in the incremental classifier in order to predict the class of the relationship. The values of the elements belong to $\{-1, 0, 1\}$. -1 represents the inequality of elements. 1 represents the equality and 0 indicates the absence of one of the descriptors. The relation R between the couple of pages p_i and p_{i+1} is defined by: $R = \{v_j : v_{j=1,\ldots,8} \in \{-1, 0, 1\}, v_{j=9} \in \{-1, 1\}\}$. Fig.5 illustrates how the relational vector R is constructed between p_i and p_{i+1}. Since there exists at least one intersection between f_1^i and f_1^{i+1} then $v_1 = 1$. Descriptors $f^i_{2,3,5,7}$ do not exist in p_i nor p_{i+1} therefore $v_{j=2,3,5,7} = 0$. Since $f_8^i < f_8^{i+1}$ then $v_8 = 1$. Finally, $v_9 = 1$ because the distance between the width of the largest blocks is almost equal.

C. Segmentation validation

Let us first start with the assumption that over-segmenting a document is preferable to fusing two documents together. Fusion implies that a document is lost since it is surely combined with the previous one, while with over-segmentation, no document is lost. Even if a document is divided into parts the recipient will receive at least one part while other parts will probably be classified correctly and sent back later. Based on this assumption, we propose a homogeneity index h to measure the segmentation accuracy. h is inversely proportional to the number of falsely fused pages (pages belonging to two different documents and fused together).



$R =$

v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9
1	0	0	1	0	1	0	1	1

Fig. 5: Example on how the relational vector R between p_i and p_{i+1} is computed

D. Documents homogeneity

Every relational vector R between two successive pages is classified based on the incremental classifier in section II as continuity, rupture. A rejection is considered when we are uncertain on whether R belongs to the continuity or rupture class.For every R we compute $Q_x = P(continuity|R) - P(rupture|R)$. If $Q_x < \delta$ we consider that we have a rejection. Rejections will only be measured on vectors where the predicted labels indicate a continuity. Every class affects the stream analysis according to the following rules:

- **Continuity**: we consider that p_i and p_{i+1} belong to the same document and we pursue the analysis to the next page.
- **Rupture**: we consider that we have a full document. All the pages between the last rupture and the new one form a document.
- **Rejection**: we consider that we have an over-segmentation. A fragment is created with the previous pages. The stream analysis continues till a rupture is reached. The result is a set of fragments.

Table.III illustrates the confusion matrix for the rejection measure. FP (False Positive) represents a *falsely* rejected relation R. An over-segmentation will take place.

TABLE III: Confusion matrix

Rejection	Predicted	Ground Truth	
True	Continuity	Rupture	***TP***
	Continuity	Continuity	***FP*** over-segmentation
False	Continuity	Rupture	***FN*** fusion
	Continuity	Continuity	***TN***

In the case of FN (False Negative), the classifier did not reject the miss-classified label. Therefore, a fusion of two pages belonging to two different documents occurs. The

lower FN, the better the segmentation. This case is risky as documents might be fused and lost. The homogeneity h is inversely proportional to FN; as h increases, FN decreases, it is determined as: $h \propto \frac{1}{FN}$

VI. Verification

Segmented pages are combined into a single document based on the method proposed by [4]. For every page p_i a bag of words is constructed by using the *tf-idf* algorithm. Then an average pooling followed by *L2-normalized* is performed to fuse all the bag of words into one representative vector u. u is represented in an *n-dimensional* space. The choice of n is based on the following experiment: 6 sets of 6000 documents were constructed by varying the size n of the bag of words. We use the k-fold cross validation in order to know which vector dimension improves the performance of the incremental classifier.

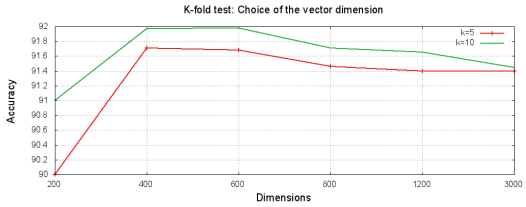


Fig. 6: Choice of the best vector dimension

By adding more dimensions as shown in Fig.6 there is little increase in the performance of the classifier. The classifier accuracy in fact starts to decrease after a certain limit. Thus we decided to set $n = 400$. The representative vector u is sent to the classification module to predict its context *(invoice, form, insurance, etc.)*. The context serves as a query to search for similar cases in the Knowledge database. The Knowledge database is simulated by a domain expert. It is only used to handle the correct labels to update the classifier [11].

VII. Results and Experiments

We propose to use 3 databases composed of real world documents. Our objective is to test our system individually on each database and on the combination of 3 of them. Databases are provided by the **ITESOFT** company.

- **database 1**: 761 documents, 1522 pages, 15 classes
- **database 2**: 1,280 documents, 5,039 pages, 23 classes
- **database 3**: 3,797 documents, 11,753 pages, 137 classes
- **database 4 (combination of all the 3 databases)**: 5,106 documents, 16,882 pages , 164 classes

A. Feature extraction

To test the stability of the feature extraction algorithm, 393 pages were randomly chosen from the 3 databases. A Ground truth was constructed by labelling manually all words in the *XML* files. The system automatically counts the number of correctly extracted features f_j on every page by comparing the expected with the ground truth labels. Fig.7 shows the stability of the proposed feature extraction algorithm based on regular expressions.

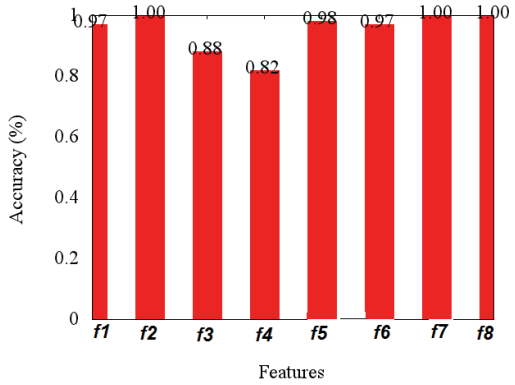


Fig. 7: Feature extraction accuracy

B. Homogeneity index

To test the performance of our proposed algorithm on the order of the documents arrival in a stream. Every database is shuffled 100 times to simulate different random streams. The homogeneity index h is computed for every shuffle. Fig.8 shows the homogeneity index h on the 4 databases with an average accuracy of $\approx 84\%$. The variation of h is normal, since the incremental classifier receives different documents with every shuffle. Even with the increase of the database size h remained generally stable.

C. Document classification

The last part of the experiments consists in measuring the classifier's accuracy in predicting the right class of the document that arrives progressively in a stream. This experiment was performed on *database 4* since it covers all the classes of documents. Fig.9 shows that the predicted accuracy over 100 shuffles is about 93%. The classification accuracy confirms that our proposed algorithm attributes correctly the class in the majority of cases. It remains stable even when changing the order of the documents in the stream. The effect of the confidence threshold δ seen in section.II on the classification is illustrated in Fig.10a. This experiment is performed on *database 4*. δ is incremented by $+0.1$ each time. The higher δ the higher the rejection rate is (see Fig.10b). Deciding to have a high rejection rate or not is a very important decision. If the number of errors needs to be minimized then the rejection rate must be increased.

VIII. Conclusion

We presented in this paper a novel algorithm for the segmentation and classification of multipage document streams. The novelty of our approach lies first in the proposed segmentation algorithm based on the ternary logic. Second, on the use of two independent modules and two types of features to make the rupture and continuity decisions. The incremental classifier showed that it is well adapted for the stream

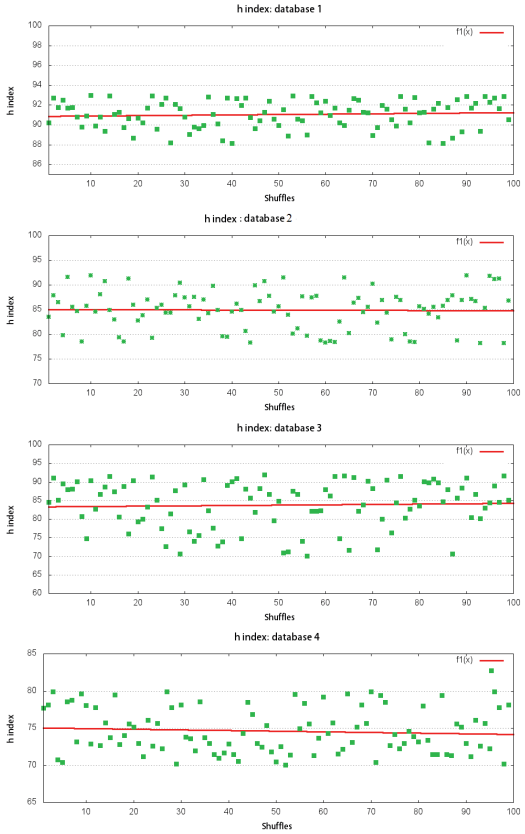


Fig. 8: Algorithm stability on the *4* databases (linear regression curve $f_1(x) = ax + b$)

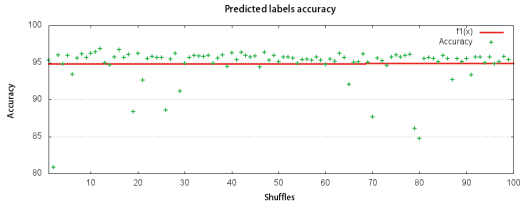


Fig. 9: Predicted labels accuracy

segmentation problem. Results showed that the segmentation features are more adapted to continuity rather than rupture situations. Therefore new rupture features should be added to the classifier. Lastly, the correction module which is simulated by a domain expert, will be automated to limit any human interaction with the system and make it more autonomous.

REFERENCES

[1] K. Collins-Thompson and R. Nickolov, *A clustering-based algorithm for automatic document separation*. Special Interest Group on Information Retrieval, pp.38-43, 2002.

[2] T. Meilender and A. Belaïd, *Segmentation of continuous document flow by a modified backward-forward algorithm*. Document Recognition and Retrieval, pp.1-10, 2009.

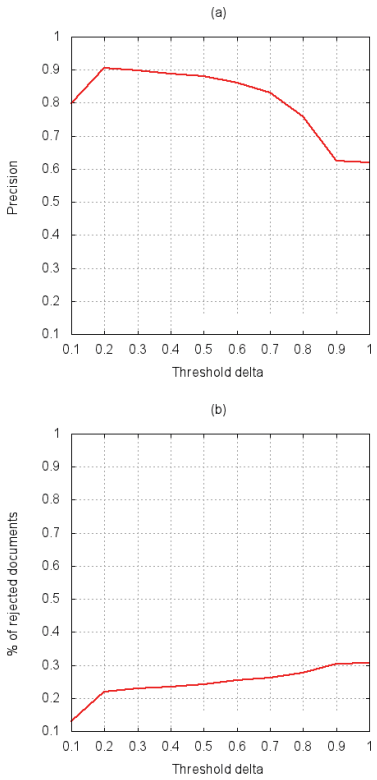


Fig. 10: Effect of δ on the precision and rejection rates

[3] M. Schmidtler and J. Amtrup, *Automatic document separation: A combination of probabilistic classification and finite-state sequence modeling*. Natural Language Processing and Text Mining, pp. 123-144, 2007.

[4] A. Gordo, M. Rusinol, D. Karatzas and A. D. Bagdanov, *Document Classication and Page Stream Segmentation for Digital Mailroom Applications*. International Conference on Document Analysis and Recognition, pp.621-625, 2013.

[5] A. Rusiñol, M. Karatzas D. Bagdanov and J. Llados ,*Multipage document retrieval by textual and visual representations*. International Conference on Pattern Recognition, pp.521-524, 2012.

[6] J. Kumar, P. Ye and D. Doermann, *Learning Document Structure for Retrieval and Classication*. International Conference on Pattern Recognition,pp.1558-1561, 2012.

[7] C. Shin and D. Doermann, *Document image retrieval based on layout structural similarity*. Internation Conference on Image Processing, pp.606-612, 2006.

[8] C. Shin,D. Doermann and A. Rosenfeld, *Classification of document pages using structure-based feature*. International Journal on Document Analysis and Recognition, pp.232-247, 2001.

[9] A. Gordo and F. Perronnin, *A bag-of-pages approach to unordered multi-page document classification*. Internation Conference on Pattern Recognition, pp.1920-1923, 2010.

[10] M. R. Bouguelia, Y. Belaid, A. Belaid, *Document image and zone classification through incremental learning*. International Conference on Image Processing, pp.4230-4234, 2013.

[11] M. R. Bouguelia, Y. Belaid, A. Belaid,*A Stream-Based Semi-Supervised Active Learning Approach for Document Classification*. International Conference on Document Recognition, pp.611-615, 2013.

[12] D. Furcy and S. Koenig, *Limited Discrepancy Beam Search*. International Joint Conference on Artificial Intelligence, pp.125-131, 2005.