

Segmentation of Continuous Document Flow by a modified Backward-Forward algorithm

Th. Meilender and A. Belaïd
University Nancy 2 - LORIA
{Thomas.Meilender, abelaid}@loria.fr

Abstract

This paper describes a segmentation method of continuous document flow. A document flow is a list of successive scanned pages, put in a production chain, representing several documents without explicit separation mark between them. To separate the documents for their recognition, it is needed to analyze the content of the successive pages and to point out the limit pages of each document. The method proposed here is similar to the variable horizon models (VHM) or multi-grams used in speech recognition. It consists in maximizing the flow likelihood knowing all the Markov Models of the constituent elements. As the calculation of this likelihood on all the flow is NP-complete, the solution consists in studying them in windows of reduced observations. The first results obtained on homogeneous flows of invoices reaches more than 75% of precision and 90% of recall.

1. Introduction

The problem of continuous flow segmentation arises in several works where it is needed to separate the successive data. This is the case in speech recognition or handwritten word recognition where the segmentation is a real dilemma. This is also the case in the industry where the production chains are fed by thousands of documents made up of scanned pages, representing similar or different documents (invoices, purchase orders, insurance forms, etc.). In all the cases, we wonder about the quantity of information to take into account to lead the operation by finding the good separators. Thus, the developers always tried to find a compromise between separation and recognition so that the solutions are not too slow.

At our knowledge the nearest approach related to this topic is the one developed by Kevyn Collins-Thompson and Radoslav Nickolov in [1] which operates by page comparison. It consists in calculating several measurements of similarity for each page of flow and then carrying out a regrouping by a dedicated algorithm. This algorithm uses thresholds to accept or reject the possible regroupings. The segmentation precision of the system reaches a level of 95%. However, the literature is more abundant concerning video plan and scene cutting where some similarities exist with document flow. Two kinds of methods exist; some of them are model-based whereas the others are model-free.

The methods requiring the construction of structure models (temporal and spatial) are based on characteristic elements of the scene such as for example the posting of a logo or the appearance of a text [2, 3].

Other approaches related to the text detection are proposed in [4, 5]. Gonsel et al. [6] suggest using the follow-up of objects (for example logos) and a method of not supervised clustering to find the structure of television news. These methods are thus based on the integration of knowledge on the field and the extraction of characteristics [7]. Within the framework of the European project CLAVIS (CLAssification of audioVISual sequences), Carrive et al. [8] described a system of temporal and terminological reasoning for the classification of audio-visual sequences, able to identify in a document any kind of structures described by a model (template). These models are defined starting from a network of constraints relating to elementary objects (plan, logos, jingles...) and must be written by an operator.

On another side, the methods which do not base on models can apply to less restricted fields by using either the images, or the sound band, or both in order to gather the similar and temporally adjacent plans. In the case of a segmentation based on the visual aspects, information on the color and the movement are generally used.

Lastly, it happens that several methods intervene in the segmentation of a scene. By way of example, Huang et al. [9] use at the same time visual and auditive indices as well as information on the movements to detect the various scenes.

The approach that we propose is the opposite of the Collins-Thompson et al. method. Whereas this one seeks to characterize and identify the ruptures, we wish to identify the documents in order to isolate the beginning and the end from it. From there, it will only remain for us to place the segmentations between the various identified documents. In this context, a stochastic model is necessary to model and ensure the recognition of the pages. It must be enough flexible to compensate the possible errors generated during the analysis. Moreover, the complexity of the methods which are associated to it should be low in order to not slow down the treatment. Our choice was focused on the Markov chain which presents these properties.

The paper is organized as follows: In section 2, the multi-gram models are presented. Then, in section 3, the Forward-Backward adaptation is proposed. Section 4 will show the application of this method on continuous document flow and section 5 will give a conclusion and some perspectives.

2. Multi-gram Models

The Multi-grams aim at modeling the dependence relationships within the continuous flow. This corresponds to our approach which leads to keep in the flow only the document series presenting a dependence relationship. In this approach, a flow is regarded as a concatenation of independent sequences of variable length. The probability of the flow is then considered as being the sum of probabilities of all the segmentations of this flow.

More formally, An N-gram is a continuation of N consecutive elements in a sequence. The principal idea of the model is to determine the probability of appearance of the element i according to its history, i.e. according to the elements which precede it. This modeling corresponds in fact to a Markov Model of order N, where the prediction takes into account N last stages.

Let us consider $F = S_0, S_1, S_i \dots S_n$, a data flow segmented according to a possible segmentation K, where $S_i = E_0, E_1 \dots E_j \dots E_m$ is one of the components stemmed from the segmentation of F by K, consists of m elements. The multi-gram model performs the likelihood L of F associated to the segmentation K as being equal to the product of the likelihood of the successive sequences.

$$L(F|S) = \prod_{t=1}^q P(s(t))$$

Now, if K is all of the possible segmentations of F, then the likelihood of F is the sum of all the possible segmentations:

$$L_{\mu gr}(F) = \sum_{S \in |S|} L(F|S)$$

Since our model is oriented towards the best possible segmentation, we can approximate the likelihood by:

$$L_{\mu gr}^*(F) = \text{Max}_{S \in |S|} L(F|S)$$

For example, if $F = "abcd"$, where a, b, c and d are elements of the flow, the likelihood estimation resembles to:

$$L_{3-\mu gr}^*(abcd) = \max \left\{ \begin{array}{l} P(a)P(bcd) \\ P(abc)P(d) \\ P(ab)P(cd) \\ P(a)P(b)P(cd) \\ P(a)P(bc)P(d) \\ P(ab)P(c)P(d) \\ P(a)P(b)P(c)P(d) \end{array} \right\}$$

Classically, with an N-gram model, the likelihood would be calculated as follows:

$$L_3 - abcd = P(a)P(b|a)P(c|ab)P(d|bc)$$

The difference between the two equations highlights the importance of the segmentation principle in the multi-gram model, then that this one does not appear in the traditional processing by an N-gram model.

3. Forward-Backward Adaptation

3.1. Traditional N-gram Model

In a traditional N-gram model, the Forward-Backward algorithm is used to obtain the likelihood of a succession of observations, knowing the Markov Model used. In Deligne and Bimbot [10], a version of this algorithm is proposed. However, it is oriented towards language model and is optimized to highlight the lexical dependences within a sentence. Our problem is different in the sense where the considered segments are independent two by two.

We try to adapt the algorithm to the case of flow document segmentation. In this case, two additional parameters are considered: 1) a set of models specialized in the recognition of the classes of elements composing the flow. Let $\Lambda = \{\lambda_0 \dots \lambda_c\}$ be this HMM set, 2) a set of possible segmentations for the processed flow, noted $S = \{S_0 \dots S_q\}$.

The goal of the Forward-Backward function is to determine the most favorable segmentation. The size of the segmentation lists depends on the number of elements composing the flow, leading to a length equal to $2^{(\text{number of element})-1}$.

The principle of the algorithm is as follows: for all the possible segmentations, the likelihood of each segment is evaluated for all the Λ models. We can first isolate the model recognizing each best segment, and from there, calculate the likelihood of the segmentation tested. Then, the segmentation having the best likelihood is retained. Hence, the principle of Forward-Backward, in the context of the data flow segmentation, is to maximize the flow likelihood knowing a set of HMMs by selecting the most adapted segmentation.

If $S = \{s_0 \dots s_z\}$ is a possible segmentation of F , then:

$$P(s|\Lambda) = \max_{0 \leq j \leq c} P(s|\lambda_j)$$

which leads to choose the more adapted model for the segment recognition. The likelihood processing of a segment knowing a model uses the traditional Forward-Backward function. We can then deduce from it the likelihood of the data flow for a given segmentation:

$$P_s(F|\Lambda) = \prod_{i=0}^z P(s_i|\Lambda)$$

The calculation of Forward-Backward leads to calculate:

$$P(F|\Lambda) = \max_{0 \leq s \leq q} P_{S_s}(F|\Lambda)$$

3.2. Sliding windows

As defined previously, the Forward-Backward algorithm tests all the possible segmentations for each flow. However, by counting these combinations, we realize that the algorithm will have a complexity of $O(2^n)$. Being given the size of the flow (more than 200 elements), it is necessary to reduce the number of cases in order to avoid the combinative explosion. The solution suggested consists in segmenting reduced windows, having a limited number of elements. For example (see Fig. 1, Case 1), for a window of 4 elements, the number of possible segmentations is 8, which decreases the algorithm complexity to $O(n)$. The method then consists to evolve along the flow by a window so as to segment it locally.

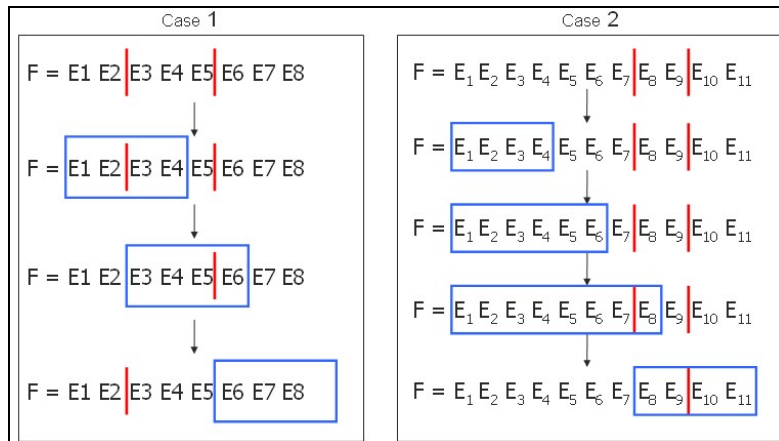


Fig. 1: Adaptation of the Backward-Forward function to the segmentation of a data flow

Then, we consider the cases where a segment consists of more than elements that a window can contain. The solution consists in increasing the window gradually until the segmentation gathers all the elements. By increasing the size in a restricted way, for example by two elements, we remain in a reduced combinatory, knowing that the first elements belong to the same segment (see Fig. 1, Case 2).

4. Application to a document flow

The corpus studied in this work is a document flow provided by ITESOPT Company. All the documents are of the same type (i.e. invoices made up of a variable number of pages) which implies a greater difficulty of the segmentation. Indeed, it is more difficult to distinguish two elements of the same type as two elements of different type. A great variability exists however, such as the language: certain invoices are in French, others are in English or German. Moreover, according to country of issue's, legal information is not the same.

The database considered consists of 356 documents distributed out of 719 pages. The database is divided into two parts: the training set (229 documents for 471 pages) and the test set (127 documents for 248 pages). Each page is provided in the form of an image file (tiff). A long and fastidious work of sorting and labeling were necessary to make the database exploitable. We simulated the arrival of an entering flow, composed of successive documents. Each page corresponds to a flow element, the objective being to reconstitute the original segmentation in documents, like that preexistent with the work of digitalization by the scanner.

4.1. Observation extraction

Each page, recognized by OCR, is described by a list of 19 observations (features). Four of them are extracted by a thesaurus method and 15 by field identification. The first features belong to a list of keywords representative of the document type and stored in a thesaurus. For example, in the case of invoices, one of the thesauruses created contains a list of terms allowing the location of the payment zone like "to pay front", "by transfer", etc. (see Fig. 2, (a)). The second observations are extracted by field identification. In this method, we associate a numerical field to a string. The search for a "total" in an invoice illustrates the method perfectly. We group together in a thesaurus the

words containing the title field such as "total", "total including all taxes", "amount", etc. Then, when one of the words is detected, a procedure controls the words to which the co-ordinates are closest, and checks if it does not find a numerical field corresponding to a predefined format which is specific to the required field. In the case of "total", the numerical field will have this type of format: "[0-9] *, [0-9] [0-9]". Several formats of fields can be defined (see Fig. 2, (b) where the amount "732.80" reveals the existence of the string "Montant TVA").

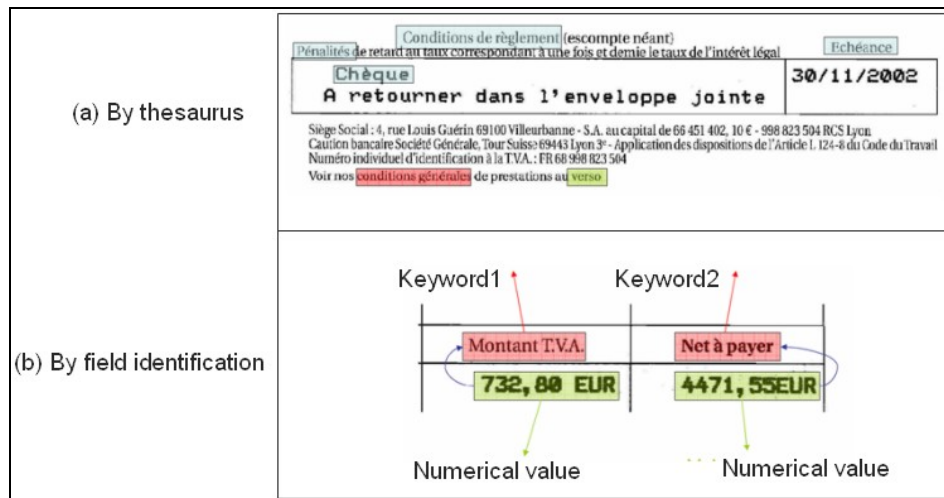


Fig. 2: Observation extraction method

Table 1 shows the word occurrences (i.e. frequency) in all the documents studied.

Table 1: Extraction rates

Field name	Extraction rate	Field name	Extraction rate
Zipcode 1	50.84%	Invoice number	50.56%
Zipcode 2	37.28%	Intermediary amount	3.38%
Phone number 1	72.03%	HT amount	76.83%
Phone number 2	15.53%	TVA amount	58.19%
Fax	51.69%	TTC amount	66.66%
SIRET	3.95%	Page number	40.39%
Order number	73.44%		

As related in the table, only a minority of fields can be extracted from a majority of documents. This doesn't mean that the fields are absent but probably that their extraction failed because of OCR errors. Moreover, OCR errors combined to error interpretation can lead to some ambiguities. The Table 2 gives the list of the 19 observations used for each page.

Table 2: Observations

Observation	Type	Description	Observation	Type	Description
O0	Bool	Numerical format of zipcode on the left side of the page	O10	Bool	Detection of TVA
O1	Bool	Numerical format of zipcode on the right side of the page	O11	Bool	Detection of a total amount
O2	Bool	Numerical format of a phone number on	O12	Int	Detection by key words of payment statements

		the left side of the page			
O3	Bool	Numerical format of phone number on the right side of the page	O13	Int	Detection by key words of a coupon of payment
O4	Bool	Numerical format of fax number on the left side of the page	O14	Bool	Detection of page number = 1
O5	Bool	Numerical number SIRET	O15	Bool	Detection of page number = 2, 3, 4,...
O6	Bool	Detection of order number	O16	Bool	Detection of page number of type 3/3
O7	Bool	Detection of invoice number	O17	Int	Detection by keywords indications announcing a following page
O8	Bool	Detection of an intermediary amount	O18	Int	Detection by keywords annexes
O9	Bool	Detection of HT amount			

4.2. Determination of observation vector

Two types of observation vectors are determined: by a vector of variable length and by a fixed length vector.

Description by a variable length vector: The goal of this approach is to find the most faithful chronological representation for each document. Each observation is added chronologically in the vector according to the place occupied in the document. We privilege here a “space-time” vision of the document, i.e. we consider the collection of information in the order where it would take place from right to left and from top to bottom. It results from it a vector whose size depends on the number of exploitable information collected (see Fig. 3 where the vector is located on the bottom).

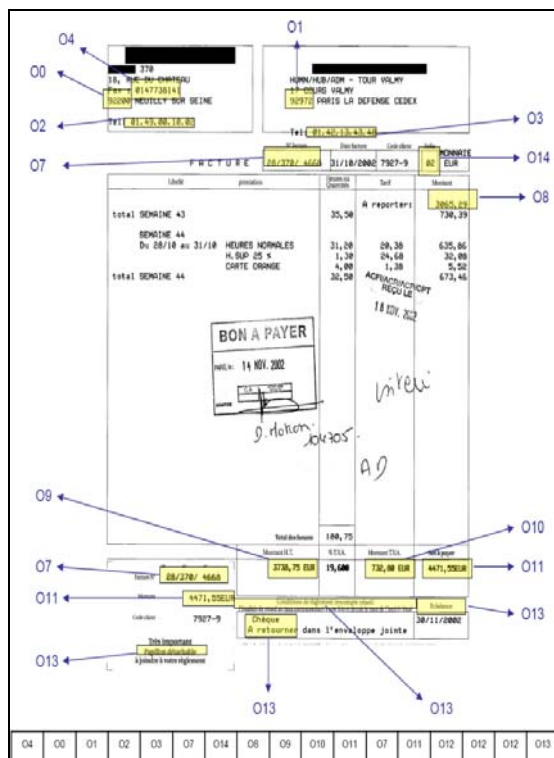


Fig. 3: Spatial-temporal invoice description

Description by a variable length vector: The goal of this approach is to create a document model which can characterize all the flow elements. The idea is to characterize similarly all the pages, independently on the information contained. Thus, we isolate a “standard” chronology from the document, more easily analyzable by the Markov models. This methodology makes it possible the system to better adapt to new documents since it will be easier to find analogies with the already known documents. For our invoices, we considered that each one can be modeled by the configuration sketched in Fig 4.

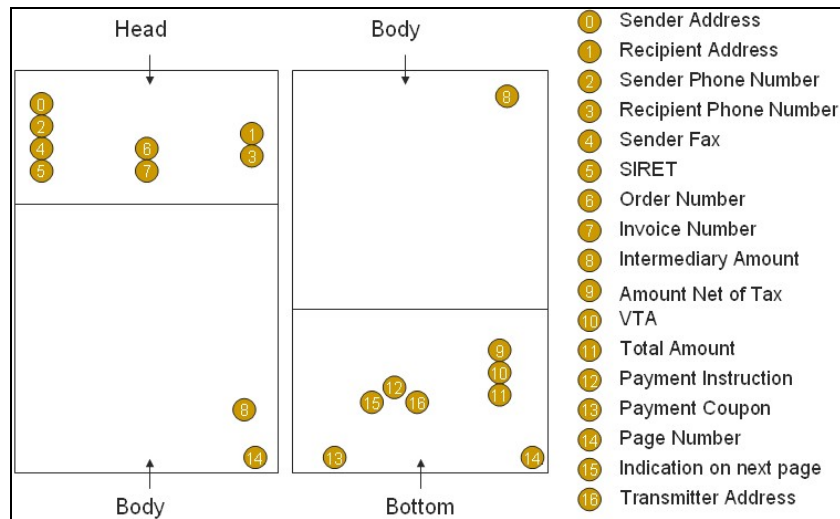


Fig 4: Description by document page modeling

This proposal enables us to build an observation vector which can inform about the quantity, the presence or the absence of information identified above. By this consideration, we fix the chronology by which the pages must be comparable by the Markov models. We obtain a vector of 19 elements having the form represented on the bottom of the Fig. 5:

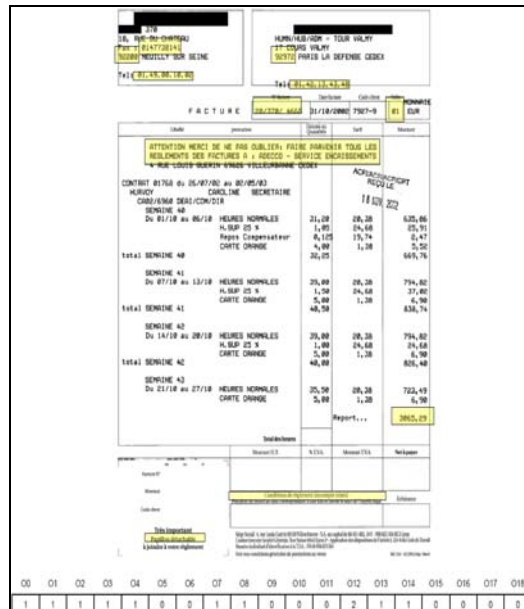


Fig. 5: Invoice representation by a vector

4.3 Training

The training corpus is divided into classes which will constitute the basis for one particular HMM. The division in classes is done in two manners: manual and automatic. The manual method is based on the general resemblance of the invoices, such as for example the identity of the transmitter, the language... We finally obtain a division into 16 classes of distinct invoices. The automatic method is usable only in the case of the vectors of fixed length. We use k-means which consists in comparing the vectors and gathering them according to a number of classes provided in parameter. The use of automatic methods allows us to open prospects for the continuation for the application use. Indeed, to apply work to new types of flows, or to adapt the models to new types of invoices, an automatic method avoids a long and fastidious sorting work.

We chose a Bakis Model which has the advantage to support the flow recognition following a fixed chronology, which corresponds to the vectors of fixed length. The choice was then confirmed empirically, similarly for the vectors of variable length. The state number was also empirically given equal to 8.

The probability distributions of each state are smoothed according to the Laplace function. The principle is to simulate the presence of a complementary element in the test corpus and to redistribute the probability density in order to avoid the impossible events and thus to compensate the absent events of the training corpus which could be met in the test corpus.

5. Recognition

The pages of the flow are described by a list of their feature vectors. The recognition process assigns their vector to the different classes thus constituted.

5.1. Evaluation method

For the evaluation, we use the precision and the recall. Precision is defined as the proportion of right elements compared to the number of found elements (correct and forgery). While recall is seen as the proportion of right elements compared to the theoretical number of elements. Consequently, a strong precision will indicate a great number of right elements and a low number of false elements. A good value of recall will indicate a low number of missing elements. Finally two measurements can be combined in the F-measurement, which will provide a measurement of total effectiveness. It is calculated in the following way:

$$F = \frac{2 * precision * recall}{precision + recall}$$

5.2. Result presentation

The following table shows the results of our experiments.

Vector	Variable length	Fixed length	Fixed length	Fixed Length	Fixed length	Fixed length	Fixed length
Sort	Manual	Manual	K-m	K-m	K-m	K-m	K-m
Classes	16	16	10	15	16	17	20
Recall	0.92	0.95	0.98	0.98	0.94	0.98	0.97
Precision	0.76	0.73	0.70	0.62	0.68	0.69	0.65
F-measure	0.83	0.83	0.82	0.76	0.79	0.81	0.78

5.3 Discussion

Concerning the recall, the best score is obtained with vectors of fixed length, sorted by a k-means in 10 classes. This score is important because it is that which corresponds more to the expectation of our industrial partner. Indeed, in the data processing sequence, an over-segmentation can be compensated in a posterior processing whereas an under-segmentation can involve a loss of data. We can also note that the recall is in all the cases higher than 90%, which shows that the theoretical segmentations are found overall. However, the results analyzed with the precision show a tendency of the system on-to segment: three solutions exceed the threshold of the 70% and only one that of the 75% (manual sorting, vector of variable length). Hence, we can observe that the two systems having reached the best scores of total effectiveness are obtained following manual sorting, which can be explained by the fact that the k-means was satisfied with local minima. It is also interesting to note that these two tests use the same number of classes, but for vectors of different nature.

6. Conclusion and future work

In this paper, we proposed and tested a new method of continuous data flow segmentation based on HMM within the framework of the multi-gram models. The first results obtained are encouraging and invite us to make certain improvements:

6.1. Concerning the HMM

Several topologies were tested and the best was retained. We can however wonder whether the model does not correspond to a local minimum. Moreover, it can be interesting to seek a method which would optimize the choice of the model according to the information contained in the training corpus. Thus, to each class an optimal HMM would correspond to it.

6.2. Concerning the observations

In our study, we base the analysis only on textual information. The consideration of perceptual information varied such as the presence of a logo, the size of the page would be an interesting prospect. It is advisable to relativize however the contribution of these indices according to their cost in term of extraction time. Indeed, the context of the application requires a fast treatment and the extraction of this type of data, when it is possible, can prove very long. This is why a study on the possibilities of extraction and the real contribution of the observation to the quality of the model is initially necessary.

6.3 Concerning the document models

The invoice model proposed previously opens a new prospect for the document processing. Indeed, the model we used is not limited to the invoice flow. We plan to extend this representation model to take into different classes of document. Accordingly, it appears judicious to use textual information in a different way. Indeed, we will seek more to characterize the vocabulary of the document. This can be done by reinforcing the extraction by thesaurus, by identifying the vocabulary specific to certain fields and thus to certain types of documents. A first approach in this direction realized on flows of forms (such as insurance forms : CIF) gave encouraging results. We can also think to create vectors characterizing the vocabulary according to the frequency of the words, as proposed in [1]. In this case, the perceptual acquisition of information takes all its importance. Indeed, all the documents can be characterized by their format, the font size used, the spacing between the lines, etc.

6.4 Concerning the contextual information

A property of a flow is to be able to provide contextual information for one of its elements. The idea would be then to compare the elements two to two, in order to know if they form parts of the same segment. This approach is diametrically opposed to ours since it will seek to characterize the ruptures, whereas we seek to recognize the segments. It is similar to the method used in [1]. It has however the advantage to highlight new sources of

observations. If one takes again the example of the invoice flow, the extraction of the sender address could be compared in the two pages and thus to provide us complementary indices on the appartenance or not to a same document.

6.5 Concerning the vector choices

The two vector construction methods offer different approaches for the representation of an element. In our application, the results obtained are finally comparable. It seems however possible to combine the two methods. The idea would consist in using the fixed length vectors for the training corpus, which allows the automatic constitution of classes, by the intermediary for example of k-means. Then, the training and the segmentation would use the vectors with variable length which offer additional information to us: the exact chronology of the document.

10. References

- [1] K. Collins-Thompson and Radoslav Nickolov, "A Clustering-Based Algorithm for Automatic Document Separation", In Proceedings of the SIGIR Workshop on Information Retrieval and OCR, (2002).
- [2] D. Swanberg, C.F. Shu, et R. Jain. "Knowledge guided parsing in video databases". In Proceedings of IS&T/SPIE Symposium on Electronic Imaging : Science and Technology, 13-24 (1993).
- [3] T. Zhang et C.C. Kuo, "Audio-guided audiovisual data segmentation, indexing and retrieval". In Proceedings of SPIE Storage and Retrieval for Image and Video Databases VII, 316-327 (1999).
- [4] M. Maybury, M. Merlino, et J. Rayson. "Segmentation, content extraction and visualization of broadcast news video using multistream analysis", In Proceedings of ACM multimedia, (1996).
- [5] A.Merlino, D.Morey, et M.Maybury. "Broadcast news navigation using story segmentation", In Proceedings of ACM Multimedia, (1997).
- [6] B. Gunsel, A. Ferman, et A. Tekalp. "Temporal video segmentation using unsupervised clustering and semantic object tracking". Journal of Electronic Imaging, 7(3), 592-604 (1998).
- [7] D. Zhong and S. F. Chang, "Structure Analysis of Sports Video Using Domain Models", IEEE International Conference on Multimedia and Expo, August 22-25, Waseda University, Tokyo, Japan, (2001).
- [8] J. Carrive, F. Pachet, et R. Ronfard. Clavis, "A temporal reasoning system for classification of audiovisual sequences". In Proceedings of Content-Based Multimedia Information Access (RIAO) Conference, Paris, France, (2000).
- [9] J. Huang, Z. Liu, et A. Rosenberg, "Automated semantic structure reconstruction and representation generation for broadcast news". In Proceedings of SPIE Storage and Retrieval for Image and Video Databases VII, 50-62 (1999).
- [10] S. Deligne and Frederic Bimbot, "Language Modeling by Variable Length Sequences: Theoretical Formulation and Evaluation of Multigrams", In Proceedings of Eurospeech, (1995).