Robust TV Stream Labelling with Conditional Random Fields

Abir Ncibi*, Emmanuelle Martienne†, Vincent Claveau‡, Guillaume Gravier‡ and Patrick Gros*
*INRIA †IRISA-Univ. of Rennes 2 ‡IRISA-CNRS
Campus de Beaulieu, F-35042 Rennes, France
Email: firstname.lastname@irisa.fr

***Abstract*—Multi-label video annotation is a challenging task and a necessary first step for further processing. In this paper, we investigate the task of labelling TV stream segments into programs or several types of breaks through machine learning. Our contribution is twofold: 1) we propose to use simple yet efficient descriptors for this labelling task, 2) we show that Conditional Random Fields (CRF) are especially suited for this task. In particular, through several experiments, we show that CRF out-perform other machine learning techniques, while requiring few training data thanks to its ability to handle the different types of sequential information lying in our data.**

***Keywords*-Conditional Random Fields; video-stream labelling; TV segmentation; robust descriptors; sequentiality.**

I. INTRODUCTION

Many digital TV channels have emerged in the recent years, making large amounts of video streams available. Yet, any new service based on these streams, such as video retrieval, information extraction, repurposing, requires, as a first step, to be able to structure the video flow into meaningful elementary units. In practice, the process of video stream structuring requires two main tasks: (1) a segmentation task that consists in detecting program boundaries, and (2), a labelling task that consists in giving each program a label describing its type or content.

Several studies [1]–[4] have already dealt with this issue. However, their main drawback is that their labelling processes chiefly rely on program information provided by the channels, on some reference databases, or on TV program guides. They all have underlined the limits of using such an external knowledge which is sometimes inaccurate or incomplete. In particular, TV guides don't contain information for small programs like commercials. Moreover, such TV guides are not always available. In this paper, to avoid this pitfall, we explore a different approach based on supervised machine learning. Of course, such an approach also requires some expert knowledge to build a training set, but we assume that this supervision is more easily available than a complete program information. More precisely, our goal is to investigate the use of a specific machine learning technique for the labelling task, namely the Conditional Random Fields (CRF), which are known to be suited to handle sequences. In that respect, our objective is manifold; we show that:
1 – CRF are efficient to induce programs labels, and out-perform other standard machine learning techniques;
2 – these good results can be obtained with few data and simple but robust descriptors;
3 – this good performance can be theoretically explained by the CRF's capability to use contextual relationships among a sequence of programs.

The remainder of this paper is organized as follows: next section is an overview of related work. In Section III, basic information about CRF and their learning algorithms are presented. Then, in Section IV, we detail our experimental settings, including the datasets used, the features and the evaluation measures. Sections V, VI and VII report the experiments we performed. Finally, we conclude in Section VIII.

II. RELATED WORK

To our knowledge, [1] are the first who proposed a complete solution for the video stream structuring problem. Their approach requires a reference database containing different kinds of breaks that are manually annotated. Breaks that repeat in the video stream are detected by matching the video stream with the breaks included into the reference database. If the video stream contains a new break that is not in the database, this new break is added to the database to update it. The main drawback of this method is its dependency to the reference database. Indeed, the latter has to be created for each channel and updated periodically to take into account all the breaks broadcasted by this channel. Another approach is proposed by [3] and consists in modelling program schedules by contextual hidden Markov models, that are able to predict all the possible schedules for a particular day. This approach gives good results in terms of precision of the prediction, but requires many annotated learning data. Another method developed by [4] uses an Inductive Logic Programming (ILP) tool to identify two classes of broadcasts: programs and breaks. The drawbacks of this approach are twofold: (1) it requires at least seven days of manually annotated programs and (2) it is not able to identify different kinds of breaks.

In this paper, we focus mainly on the labelling task. We suppose that the video stream has been divided, manually or automatically, into sequences of video segments. We propose a robust approach that uses CRF to label all the resulting

 ISBN: 978-1-61208-265-3

video segments. The highlights of our method are: (1) each segment is described with robust descriptors that are very easy to compute, and (2) the use of CRF allows for building an efficient model that predicts the types of the segments by taking into account the sequentiality of the data and the relationships between neighbouring segments.

CRF have been successfully applied to text processing, such as part-of-speech tagging [5], [6] or shallow text parsing [7]. They have also been used in video processing for detecting semantic events [8], [9] or identifying players in sports videos [10]. For all these tasks, CRF proved high efficiency and outperformed other probabilistic models, especially generative models like Hidden Markov Models (HMM) [11] and other discriminant models like Maximum Entropy Markov Model (MEMM) [12].

III. CONDITIONAL RANDOM FIELDS: BASIC CONCEPTS AND RELEVANT ALGORITHMS

A. Basic concepts

Conditional random fields [12] are undirected graphical models which aim at modeling a probability distribution of annotations y conditioned on known observations x based on labelled examples. CRF are defined as follows: we assume $G(V, E)$ an undirected graph (graph of independence) where V are vertices of the graph and E are edges of the graph. X and Y are two random fields over respectively the set of observations and the associated set of labels. For each vertex $v \in V$, it exists a random variable Y_v in Y. (X,Y) is called a conditional random field when each random variable Y_v depends only on observations X and its neighbours in the graph G. Based on this condition and according to the fundamental theorem of random fields (Hammersley and Clifford, 1971), the conditional probability of a sequence of annotations y given a sequence of observations x is written in terms of potential functions ψ_c over all cliques of the graph G:

$$p(y|x) = \frac{1}{Z(x)} \prod_{c \in \mathcal{C}} \psi_c(y_c, x) \tag{1}$$

where:

- $\mathcal{C}$ is the set of all cliques of the graph G (completely connected subgraphs).
- y_c are configurations of random variables over vertices of the clique c.
- $Z(x)$ is a normalization factor.

B. Linear-chain CRF

The main use of CRF in the literature, mainly in natural language processing, is labelling sequences. In this case, the graph of independence G is a first-order linear chain (as the one shown on Fig. 1).

In this graph:

- cliques are adjacent edges and vertices of the graph;

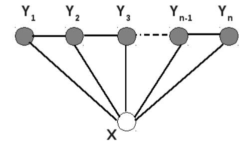

Figure 1. Graphical representation of a sequential CRF

- each label depends only on the previous and the next labels and the entire observations sequence x.

For linear CRF [12], the potential function ψ_c can be written as an exponential of weighted functions over the two types of cliques of the graph G as follows:

$$P(y|x) = \frac{1}{Z(x)} exp \left(\sum_{k=1}^{k_1} \sum_{i}^{n} \lambda_k f_k(y_i, x) + \sum_{k=1}^{k_2} \sum_{i}^{n} \mu_k g_k(y_{i-1}, y_i, x) \right) \tag{2}$$

where:

- $$Z(x) = \sum_{y} exp \left(\sum_{k=1}^{k_1} \sum_{i}^{n} \lambda_k f_k(y_i, x) + \sum_{k=1}^{k_2} \sum_{i} \mu_k g_k(y_{i-1}, y_i, x) \right) \tag{3}$$
- f and g are called *features functions*. The f functions characterize local relations in terms of labels and link the current label at position i to the sequence of observations x; the g functions describe transitions between the graph vertices (states) and are defined for each pair of labels (or states) at position i and $i-1$ and the sequence of observations.
- k_1, k_2 and n are respectively: number of features functions f, number of features functions g and the size of the sequence of labels to be predicted.

Functions in f and g are generally binary functions which show the occurrences of particular combinations of label(s) and observation(s). These functions are fixed by the user, they reflect the knowledge of the user on the application field. Each function is applied to all the positions of the sequence. For instance, let's define $f(x_i, y_i)$ which relates the current observation x_i to its current label y_i. If we apply this function to all couples of labels and observations in the sequence, it will generate $|x| \times |y|$ functions features. Let's consider the sequence of observations $x = (15s, 10m, 10s, 1h)$, where each observation is the duration of the corresponding program, and its associated sequence of labels $y = (commercial, trailer, commercial, program)$. We also suppose that we fix two functions $f(x_i, y_i)$ and $g(y_i, y_{i-1})$. At position $i = 3$, the following feature functions are generated:

$$f(x_i, y_i) = \begin{cases} 1 & \text{if } x_i = 10s \text{ and } y_i = commercial \\ 0 & \text{else} \end{cases}$$

$$g(y_i, y_{i-1}) = \begin{cases} 1 & \text{if } y_i = commercial \text{ and } y_{i-1} = trailer \\ 0 & \text{else} \end{cases}$$

Features functions are associated with weights λ_k and μ_k that estimate the importance of information given by each feature function.

The conditional nature of CRF allows for relaxing the assumption of the conditional independence of observations fixed in the HMM, and allows for neighbourhood interactions among the observed data. CRF also avoid the *label bias* problem met with the HMM (or extensions like MEMM). This problem is caused by the fact that the probability mass received by y_{t-1} *must be* transmitted to y_t (at time t) regardless the corresponding observation x_t (for the interested reader, a good illustration of the label bias problem is presented by [12]). CRF are not impacted by such considerations since the way adjacent pairs y_t and y_{t-1} influence each other is not directed and is determined by input features x.

C. Learning and inference with CRF

Learning CRF models consists in estimating the vector of parameters $\theta = (\lambda_1, \lambda_2, ..., \lambda_{k_1}, \mu_1, \mu_2, ..., \mu_{k_2})$ given a training set $D = (x^{(i)}, y^{(i)})_{i=1}^{i=N}$ which maximizes the log-likelihood of the model:

$$L_\theta = \sum_{i=1}^{N} log(p_\theta(y^{(i)}|x^{(i)})) \quad (4)$$

This function is concave, guaranteeing convergence to the global maximum. This optimization can be resolved by traditional iterative scaling learning algorithms such as the Improved Iterative Scaling (IIS) algorithm [13] but it has been proved that the Limited memory BFGS (L-BFGS) quasi-Newton method [14] converges much faster to estimate the parameters θ. The advantage of L-BFGS is that it avoids the explicit estimation of the Hessian matrix of the log-likelihood by building up an approximation of it, using successive evaluations of the gradient.

After this step of training, applying the CRF consists in finding the most probable sequence of labels y^* given a sequence of observations x:

$$y^* = \arg\max_y p_\theta(y|x) \quad (5)$$

As for other stochastic methods, y^* is generally obtained with a Viterbi algorithm, which calculates the marginal probability of states at each position of the sequence, using a dynamic programming procedure.

IV. Experimental settings

In this section, we present the data we used for the experiments that were conducted to evaluate CRF, as well as other machine learning algorithms, on video stream labelling.

A. Data

In our experiments, we used a TV stream containing three weeks of broadcasts. Within this stream, each segment was manually identified and given a label corresponding to its type: program or break. Four additional types have also been used to distinguish between different kinds of breaks: trailer, commercial, sponsorships and jingle. Two datasets were produced from this stream, each dataset resulting from the application of a particular segmentation method:

- A manual segmentation method which identifies precisely the beginning and the end of each broadcast. This segmentation will be useful for evaluating the relevance of CRF on the labelling of a perfectly segmented video stream. In this case, a video segment is equivalent to a program (movie, TV serie, talk-show, etc) or a break (commercial, trailer...). 7,591 video segments were extracted using this manual segmentation method.
- An automatic segmentation method that aims at evaluating CRF for the labelling in a more realistic setting. The automatic segmentation method we used is based on the detection of repeated segments [15]. Applied to TV streams, this method tends to over-segment the stream: 48,544 video segments were detected. By examining the result of the segmentation, we observe that each broadcast (corresponding to a unique segment with the manual segmentation method) is divided into several segments, each segment having a short duration.

The distribution of the segments over the different types, inside both datasets, is shown on Table I.

Table I
Distribution of the segments over the different types

Label	Manual segmentation	Automatic segmentation
Program	1,506	22,557
Trailer	1,290	4,075
Commercial	1,050	18,089
Sponsorship	1,714	2,201
Jingle	2,031	1,622
Total	7,591	48,544

B. Descriptors

Within both datasets, each video segment is described by three descriptors:

- its duration: we distinguish between ten possible values, each value being an interval:[0-15s[, [15s-30s[, [30s-45s[, [45s-1min[, [1m-15m[, [15m-30m[, [30m-1h[, [1h-2h[, [2h-4h[.
- the moment in the week it was broadcasted: business day, off-day or weekend.

- the period in the day it was broadcasted: morning, noon, afternoon, evening, night.

These features are robust since they are very easy to compute, and they do not depend on the quality of image or sound signal of the stream. Some examples of segments with these features and their labels are shown in Table II.

Table II
EXAMPLES OF SEGMENTS WITHIN THE DATASETS, WITH THEIR FEATURES AND LABELS

Segment	Moment in the week	Period in the day	Duration	Label (class)
Seg_{15}	Business day	morning	[10s,15s[	commercial
Seg_{16}	Business day	morning	[0s,10s[	trailer
Seg_{17}	Business day	morning	[0s,10s[	jingle
Seg_{18}	Business day	afternoon	[15min,30min[	program
Seg_{19}	Business day	afternoon	[10s,15s[	commercial

C. Labelling tasks and evaluation measures

For all the experiments, the first two weeks of a dataset were used to train and construct the labelling model, and the last third week to test and evaluate this model. Two kinds of labelling tasks have been considered:

- a binary labelling task in which a segment is either a program or a break, i.e., there is no distinction between different kinds of breaks;
- a multiple labelling task in which a distinction is made between different kinds of breaks. Consequently, five labels are used: program, commercial, sponsorship, trailer or jingle.

In the experiments reported in the next sections, the performance is evaluated on the test sequences by comparing the labels produced by the technique with those from the ground-truth. Different evaluation measures are used. As a global measure, we compute the accuracy rate, that is, the proportion of correctly labelled segments in the test streams. For each label, we also evaluate the recall, precision and f-score, and we then compute a weighted average over the labels (weighted according to the amount of segments of each class). Note that the weighted average recall is equivalent to the accuracy rate that we use as a global measure.

D. Video stream labelling with CRFs

To use efficiently CRF, we consider that a sequence groups together all the segments of a day of broadcasting. We have 15 sequences, i.e., the first two weeks for learning the labelling model and 8 sequences, i.e., the last third week, for testing the model. In a sequence, observations are the vectors of descriptors and labels are the types of the segments.

To learn the labelling model, appropriate features functions must be chosen to express the dependencies that may exist in a sequence between observations or labels. In our experiments, we used the tool CRF++[1]. In this tool, feature functions are defined in a *template file* where:

- feature functions f are equivalent to *unigram templates* that describe relationships between the current label and the observations in the sequence (see Section III-A);
- feature functions g are equivalent to *bigram templates* that describe only the relationships between two successive labels (see Section III-A).

For each dataset, an appropriate template file which defines the f and g functions was created:

- *Manual segmented stream dataset*: for this dataset, we used feature functions which run over a window of eight neighbours; four before and four after the current observation.
- *Automatic segmented stream dataset*: for this dataset, feature functions are more complicated. We used also a window of eight neighbors but more combinations of previous, current and following observations were taken into account. For this dataset, we used only local feature functions f (unigrams) to avoid the over-fitting.

In Section VII, we presents some results obtained with different templates. Each experiment was performed on both datasets. To highlight the efficiency of CRFs in sequential data labelling, we compare the results obtained by CRFs with the results obtained by different non sequential classification methods (SVM, Naive bayes, Random Forest) for the same labelling task. For each method, two settings are presented. The first one uses a naive description in which only the current observation is considered (noted as *simple* hereafter). The second one (noted as *contextual* hereafter) takes into account the context of the observations by adding the descriptors of the surrounding observations in the description of the current segment. Different sizes of contexts have been tested; here, we report the ones yielding the best results, that is when considering the two previous and the two next segments. We also compare CRFs to HMMs to study the impact of the label context which is also taken in account by CRFs. To the contrary of CRF, HMM only take into account the current observation of the segment to be labelled. To complete this comparison, we also indicate the results of two baselines:

- *Baseline1* where only the most frequent label is predicted;
- *Baseline2* which uses a features function which considers only the duration of the current segment to be labelled. We choose this baseline because duration is the most discriminant descriptor.

Again, the experiments are performed on both manually and automatically segmented datasets.

[1] http://crfpp.googlecode.com/svn/trunk/doc/index.html

V. Results on the manual dataset

This section is dedicated to the results obtained with the manually segmented stream. In the first part, we summarize the results obtained by CRF and other classification methods. In the second part, we focus on the detailed results obtained by CRF on binary and multiple labelling.

A. Global results on the manual dataset

Results are presented on Tables III and IV. The best results for SVM, reported hereafter, were obtained with a RBF (Radial Basic Function) kernel where γ is set to 0.1. For the results, we attribute to each label a weight that is proportional to its frequency in the learning dataset. Then, we calculate weighted averages of recalls, precisions and F-scores for each predicted label.

Table III
Performance for the binary labelling task with the manual dataset, using CRF and various classification methods

	Accuracy - Recall (%)	Precision (%)	F-score (%)
CRF	**95.66**	**95.63**	**95.63**
HMM	88.79	90.2	89.2
SVM simple	86.3	85.4	85.2
N.Bayes simple	86.6	87	86.8
R.Forest simple	87.1	88.3	87.5
SVM contextual	94.9	94.8	94.8
N.Bayes contextual	89.4	91.0	89.9
R.Forest contextual	94.9	94.8	94.8
Baseline1	79.48	63.16	70.39
Baseline2	87.37	86.66	86.53

From these results, several points are noteworthy. First, the task of binary labelling seems easy enough to yield high score with the baseline techniques. Secondly, the difference between the simple and contextual settings of the usual machine learning algorithms underlines the importance of taking the context of the current observation into account as it is naturally done with CRF. Thirdly, the label context, naturally taken into account by HMM and CRF, seems also beneficial for the performance.

Table IV
Performance for the multiple labelling task with the manual dataset, using CRF and various classification methods

	Accuracy - Recall (%)	Precision (%)	F-score (%)
CRF	**84.08**	**85.13**	**84.54**
HMM	74.52	53.82	49.22
SVM	59.5	64.6	56.5
N.Bayes	59.6	62.8	56.5
R.Forest	58.7	61.3	55
SVM contextual	76.1	76.8	75.8
N.Bayes contextual	68.6	69.6	68.8
R.Forest contextual	74.9	75.8	74.6
Baseline1	27.36	7.49	11.76
Baseline2	64.3	66.2	64.8

The multiple labelling task is more difficult, resulting in lower performance. Here again, the importance of taking the context of the current observation into account appears clearly. As it is suggested by the HMM results, the context of the label is not enough to cope with these more complex data. Yet, the CRF model, which takes both contexts into account yields the better results and outperforms any other technique. Finally, these results highlight the two following interesting points:

- using features functions, CRF are the most competitive method for the task of video sequence labelling;
- robust descriptors are discriminant enough to label the manual dataset.

B. CRF results on the manual dataset

In order to analyse the errors, we report detailed results for the CRF in Tables V and VI.

Table V
Detailed performance for the binary labelling task with the manual dataset using CRF

	Number of segments	Recall (%)	Precision (%)	F-score (%)
Inter-Program	1.184	97.52	97.03	97.27
Program	564	88.47	90.23	89.34
Weighted average		95.66	95.63	95.65

CRF have interesting results in binary labelling: both programs and breaks are identified by the learned model. Breaks are better identified than programs because they are more numerous in the learning dataset and are characterized by their short duration.

Table VI
Detailed performance for the multiple labelling task with the manual dataset using CRF

	Number of segments	Recall (%)	Precision (%)	F-score (%)
Program	564	88	92.31	90.10
Trailer	381	88.47	90.23	89.34
Jingle	752	85.37	81.87	83.53
Commercial	309	87.37	82.56	84.9
Sponsorship	742	76.17	81.43	78.71
Weighted average		84.08	85.13	84.54

In multiple labelling, CRF are still able to predict labels with a F-score equal to 84.54%. Programs are the best predicted class with 92.31% precision and 90.10% F-score.

VI. Results on the automatic dataset

The automatic segmentation technique used to produce what we refer as the automatic dataset tends to over-segment the stream, as programs or breaks are generally divided into several segments. For the labelling task, these multiple segments belonging to one broadcast, have to get the same label. This section follows the same structure than the previous one: we start by presenting the global results of

 ISBN: 978-1-61208-265-3

the different machine learning methods, before giving more detailed results about the CRF. For all these experiments, the best results with SVM were obtained with a linear kernel.

A. Global results on the automatic dataset

Results are shown in Tables VII and VIII.

Table VII
PERFORMANCE FOR THE BINARY LABELLING TASK WITH THE AUTOMATIC DATASET, USING CRF AND VARIOUS CLASSIFICATION METHODS

	Accuracy - Recall (%)	Precision (%)	F-score (%)
CRF	**69.54**	72.94	**67.9**
HMM	62.7	**73.37**	56.8
SVM simple	57.9	57.8	57.7
N. Bayes simple	57.7	57.7	57.7
R. Forest simple	58.7	58.7	58.7
SVM contextual	64.7	68.4	61.3
N. Bayes contextual	63.9	65.4	61.7
R. Forest contextual	63.4	65.7	61.7
Baseline 1	52.19	27.24	35.79
Baseline 2	54.29	55.16	53.77

Table VIII
PERFORMANCE FOR THE MULTIPLE LABELLING TASK WITH THE AUTOMATIC DATASET, USING CRF AND WITH VARIOUS CLASSIFICATION METHODS

	Accuracy - Recall (%)	Precision (%)	F-score (%)
CRF	**57**	51.25	**52.45**
HMM	37.65	**55.4**	37.32
SVM simple	46.4	36.9	40.2
N. Bayes simple	46.7	39.5	41.7
R. Forest simple	47.5	40.8	42.5
SVM contextual	50.6	45.6	45.9
N. Bayes contextual	51.0	46.3	48.2
R. Forest contextual	51.7	47.8	47.8
Baseline 1	47.8	22.85	30.92
Baseline 2	47.32	38	41.26

Several facts are worth noting. First, in the binary labelling task as in the multiple labelling one, CRF still provide the best performance (in terms of Accuracy and F-scores) compared to other methods. One other interesting point is that all the methods yield lower results than for the manual dataset, with about a 30% F-score loss. This unsurprising result can be explained by the over-segmentation that resulted from the use of an automatic segmentation tool.

B. CRF detailed results on the automatic dataset

Table IX
DETAILED PERFORMANCE ON THE BINARY LABELLING TASK WITH THE AUTOMATIC DATASET USING CRF

	Number of segments	Recall (%)	Precision (%)	F-score (%)
Break	9,198	64.97	90.34	75.58
Program	8,426	81.63	46.83	59.52
Weighted average		72.94	69.54	67.9

Table X
DETAILED PERFORMANCE ON THE MULTIPLE LABELLING WITH THE AUTOMATIC DATASET USING CRF

	Number of segments	Recall (%)	Precision (%)	F-score (%)
Program	8426	64.61	67.17	65.87
Trailer	1459	1.23	10.17	2.19
Jingle	635	0.00	0.00	0.00
Commercial	6149	74.37	49.26	59.27
Sponsorship	945	0.94	20.45	1.80
Weighted average		51.25	56.99	52.45

As for the manual dataset, we provide detailed results of the CRF performance in Tables IX, X and XI. In multiple labelling of the automatic dataset, CRF provide the highest F-score in average (52.45%), even if there is a high confusion between labels as shown on the confusion matrix (see Table XI). Commercials and programs are the best recognized by CRFs. Other broadcasts are difficult to be correctly labelled, especially jingles and sponsorships.

Table XI
MULTIPLE LABELLING OF THE AUTOMATIC DATASET USING CRF - CONFUSION MATRIX

Real \ Predicted	Program	Trailer	Jingle	Commercial	Sponsorship
Program	5444	115	8	2836	23
Trailer	562	18	0	876	4
Jingle	204	4	0	426	2
Commercial	1529	38	4	4573	6
Sponsorship	370	2	1	573	9

We note a high confusion between the following labels (see Table XI):

- jingle and commercial: more than 50% of jingles are labelled as commercials and the remainder as programs;
- trailer and commercial: more than 50% of trailers are labelled as commercials and the remainder as programs;
- sponsorship and commercial: more than 50% of sponsorships are labelled as commercials and the remainder as programs;
- commercial and program: almost 25% of commercials are labelled as programs and almost 30% of programs are labelled as commercials.

These results highlight the fact that the descriptors used to describe the segments are not discriminant enough to separate many successive segments.

VII. EXPLORING THE EFFICIENCY OF CRF

Results obtained in the previous experiments show that CRFs are better suited to our labelling tasks than other usual machine learning techniques. In this section, two related issues regarding this good performance are explored. We first shed light on the importance of taking into account the sequential nature of our data, and how this is done, at different levels in CRFs. As the supervision task is tedious and costly, we then examine how CRF deal with different training set sizes.

A. About sequentiality in CRF

Four experiments were conducted in order to shed light on the ability of CRF to take into account the sequential nature of the data. This is simply done by using different template files defining the model. Here are the different settings used for these experiments:

- *CRF-all*: this template indicates that the CRF uses a) information about the current observation as well as the four before and the four next ones (corresponding to features function $f(y_i, x_{i-4}, ..., x_i, ..., x_{i+4})$), and b) information about the neighbouring labels, called bigram template, corresponding to feature functions $g(y_i, y_{i-1})$.
- *CRF-CO*: here, we use a) an unigram template which considers only the current observation and its associated label (corresponding to features function $f(y_i, x_i)$) and b) a bigram template; so finally:

$$P(y|x) = \frac{1}{Z(x)} exp\left(\sum_{k=1}^{k_1}\sum_{i}\lambda_k f_k(y_i, x_i) + \sum_{k=1}^{k_2}\sum_{i}\mu_k g_k(y_{i-1}, y_i)\right) \quad (6)$$

  In terms of information taken into account, this formulation can be compared to the HMM one.
- *CRF-nonB*: this template is similar to CRF-all, without the bigram template.

$$P(y|x) = \frac{1}{Z(x)} exp\left(\sum_{k=1}^{k_1}\sum_{i}\lambda_k f_k(y_i, x_{i-4}, ..., x_{i+4})\right) \quad (7)$$

  This formulation can be compared to the *contextual* setting of the standard machine learning algorithms.
- *CRF-CO-nonB*: this last template only includes an unigram template which considers the current observation (corresponding to features function $f(y_i, x_i)$). This formulation can be compared to the *simple* setting of the standard machine learning algorithms.

These different CRF versions are tested on the manual dataset on the multiple label task. Table XII presents the results they obtain (report to Table IV for other methods' performance). From these experiments, one can assess the importance of the two types of sequential information taken into account in CRF. Indeed, both the label dependency and the neighbouring observations help to yield the best results. On this particular task, the latter has a greater impact than the former. It is interesting to compare the results of the CRF-CO and HMM since they both exploit the same information. Yet, the CRF clearly outperforms HMM thanks to its undirected representation of the label dependency preventing any label bias problem (cf. section III-B). As expected, CRF-nonB yields similar results to those of the *contextual* setting of the SVM, Naive Bayes or Random Forests. Similarly, the CRF-CO-nonB, whose prediction only relies on the current observation, is comparable to standard machine learning techniques such as SVM, or Random Forests with the simple setting and thus also obtains similar results.

Table XII
PERFORMANCE FOR THE MULTIPLE LABELLING TASK WITH THE MANUAL DATASET, USING CRF AND VARIOUS CLASSIFICATION METHODS

	Accuracy - Recall (%)	Precision (%)	F-score (%)
CRF-all	**84.08**	**85.13**	**84.54**
CRF-nonB	78	78.43	78.33
CRF-CO	66.27	69	66.79
CRF-CO-nonB	58.7	61.27	55.09

B. Training set size

As it has been said before, due to the cost of supervision, it is interesting to examine how the performance of the labelling techniques are dependent on the training set size. For this experiment, we adopt the most difficult setting: multiple labelling with the automatic dataset. At every learning step, we add a new sequence of segments broadcasted on the same day to the training set, learn the CRF parameters, and apply this CRF on the test set. To prevent any bias, the sequences are randomly selected, and the results are averaged over several runs. The results are shown in Fig. 2.

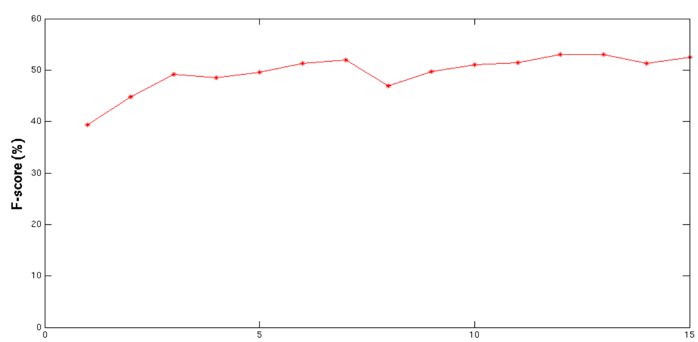

Figure 2. Results of CRF on multiple labelling of the automatic dataset, using different sizes of the learning dataset

We note that even with a small number of sequences (3 sequences), CRF have a F-score near of its optimum. This efficiency of CRF could be explained by the advantage of CRF compared to other probabilistic graphical models that do not account for local conditional probabilities, so being free of biased estimations of these quantities when too few labeled data are available. Instead, feature functions weights account for positive as well as negative contributions of the observations to the labels. This result is interesting when compared with existing methods requiring extensive annotated data such as [4].

VIII. Conclusion and further work

In this paper, we applied CRF to the labelling of a segmented TV stream where video segments are described with robust descriptors. These descriptors can be computed easily because they do not depend on the signal quality and do not require knowledge in the field of signal or image. In addition, these descriptors don't depend on a specific channel nor on the period in the year. The TV stream was segmented with two different segmentation processes, each process leading to a specific dataset: manual and automatic. Our goal was to identify five kinds of broadcasts in each dataset. We obtained interesting results on the manual dataset where the precision and the recall were up to 90%. Results are lower on the automatic dataset, especially in multiple labelling where we noticed many confusions between labels. Nevertheless, CRF's results exceed those of other classification methods such as Hidden Markov Models, which is also a probabilistic graph-based model. Indeed, the CRF's capability to handle the sequential context between video segments makes it possible to separate different kinds of programs and breaks, even when they are described with very simple features. Of course, this approach chiefly relies on the quality of the stream pre-processing steps. Dealing with the automatically segmented data is thus more challenging, especially for the multiple labelling task, which leads to high confusion between certain labels (commercial vs. jingle, commercial vs. and sponsorship...). This weakness can be explained by the over-segmentation of the automatic dataset: broadcasts are divided into many consecutive segments that features are not informative enough to discriminate.

Different perspectives are foreseen for this work. To improve our results on the multiple labelling task, especially for the automatically segmented dataset, we plan to investigate the use of content-based features, namely audio features that are specific to some kinds of breaks (for instance, there is no music and no speech in jingles). Another challenge is also to reduce the need for already labelled data in the building of the model. To achieve this goal, we plan to introduce unlabelled data and to explore using active learning strategies to induce CRF.

References

[1] X. Naturel, G. Gravier, and P. Gros, "Fast structuring of large television streams using program guides," in *4th International Workshop on Adaptive Multimedia Retrieval, AMR'06*, 2006.

[2] X. Naturel and P. Gros, "Detecting repeats for video structuring," *Multimedia Tools and Applications*, vol. 38, no. 2, pp. 233–252, 2008.

[3] J.-P. Poli, "An automatic television stream structuring system for television archives holders," *Multimedia systems*, vol. 38, no. 2, pp. 255–275, November 2008.

[4] G. Manson and S.-A. Berrani, "An inductive logic programming-based approach for TV stream segment classification," in *IEEE International Symposium on Multimedia, ISM'08*, Berkeley, Californie, USA, December 2008.

[5] A. Pranjal, R. Delip, and R. Balaraman, "Part of speech tagging and chunking with hmm and crf," in *NLP Association of India (NLPAI) Machine Learning Contest*, 2006.

[6] M. Constant, I. Tellier, D. Duchier, Y. Dupont, A. Sigogne, and S. Billot, "Intégrer des connaissances liguistiques dans un CRF : Application à l'apprentissage d'un segmenteur-étiqueteur du français," in *Traitement Automatique du Langage Naturel (TALN'11)*, 2011.

[7] F. Sha and F. Pereira, "Shallow parsing with conditional random fields," in *Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, ser. NAACL '03, 2003.

[8] T. Wang, J. Li, Q. Diao, Y. Z. Wei Hu, and C. Dulong, "Semantic event detection using conditional random fields," in *Computer Vision and Pattern Recognition Workshop (CVPRW '06)*, 2006.

[9] N. Zhang, L.-Y. Duan, Q. Huang, L. Li, W. Gao, and L. Guan, "Automatic video genre categorization and event detection techniques on large-scale sports data," in *Conference of the Center for Advanced Studies on Collaborative Research (CASCON'10)*, 2010.

[10] W.-L. Lu, J.-A. Ting, K. P. Murphy, and J. J. Little, "Identifying players in broadcast sports videos using conditional random fields," in *Computer Vision and Pattern Recognition (CVPR '11)*, 2011.

[11] L. R. Rabiner and B. H. Juang, "An introduction to hidden Markov models," in *IEEE ASSP Magazine*, 1986.

[12] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *International Conference on Machine Learning (ICML)*, 2001.

[13] S. Della Pietra, V. Della Pietra, and J. Lafferty, "Inducing features of random fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 4, pp. 380–393, 1997.

[14] N. N. Schraudolph, J. Yu, and S. Günter, "A stochastic quasi-Newton method for online convex optimization," in *11th International Conference on Artificial Intelligence and Statistics*, ser. Workshop and Conference Proceedings, vol. 2, San Juan, Puerto Rico, 2007, pp. 436–443.

[15] Z. A. A. Ibrahim and P. Gros, "TV stream structuring," *ISRN Journal on Signal Processing*, vol. 2011, no. 1, April 2011.

 ISBN: 978-1-61208-265-3