# An Integrated System for Accessing the Digital Library of the Parliament of Andalusia: Segmentation, Annotation and Retrieval of Transcriptions and Videos

Luis M. de Campos, Juan M. Fernández-Luna, Juan F. Huete and Carlos J. Martín-Dancausa

Departamento de Ciencias de la Computación e Inteligencia Artificial, E.T.S.I. Informática y de Telecomunicación, Universidad de Granada, C.P. 18071, Granada, Spain
{lci, jmfluna, jhg, cmdanca}@decsai.ugr.es

**Abstract.** In this paper, an integrated system for searching the official documents published by the Parliament of Andalusia is presented. It uses the internal structure of these documents in order to offer not only complete documents but parts of them given a query. Additionally, as the sessions of the Parliament are recorded in video, jointly to the text, the system could return the associated pieces of video to the retrieved elements. To be able to offer this service, several tools must be developed: PDF converters, video segmentation and annotation tools and a search engine, all of them with their corresponding graphic interfaces for interacting with the user. This paper describes the elements which comprises it.

## 1 Introduction

The Parliament of Andalusia was established in 1982. From that moment, this institution generates a group of electronic documents in PDF format called session diaries and the official gazettes, published in the www.parlamentodeandalucia.es site. Moreover, the sessions are recorded in video, so additionally to the transcriptions, the digital library of the Parliament is complemented with the videos.

In the session diaries, and therefore, in the videos, we can find all the participations of the members of parliament, and also all the agreements achieved in the plenary sessions of the Permanent and Commission Delegation passing laws or celebrating informative sessions with members of the regional Government.

If we take into account that each session celebrated in the parliament presents a very well defined structure, as well as the fact that each document contains an exact replica of its corresponding session, the content of each PDF is organized according to a strict and rich structure that may be useful in terms of retrieval.

In the field of Information Retrieval (IR) [1], when the retrieval mechanism is able to use the structured information contained in the documents, we are dealing with so-called structured IR [2]. Then, the internal organization of the documents is used to give back the user, instead of a whole relevant document, only those parts of them which are relevant. This means an important saving of user time.

Thanks to the internal organization of the text from session diaries and the official gazettes, the legislative collection of the Parliament of Andalusia could be studied from

a structured IR perspective. But also the videos, or the pieces of them associated to the document units could also be delivered to the user, so she/he could read the text or watch the video. This would be an added value for a search engine accessing this digital library. In this paper, we briefly present the model in which this search engine is based on, as well as the user interfaces of the search application.

But the infrastructure needed to reach this objective is quite complex, as text and video must be processed properly. First of all, the collection of PDF documents must be converted into XML, so the internal structure could be used by a search engine. A second stage is the processing of the videos, because a link must be established between a document, and its parts, and the video.of the same session. Then there is a task of synchronizing the text contained in the session diary and the corresponding video. The video is partitioned in segments of similar content, detecting the boundaries. When there is a change of camera, a new segment is created. Finally, these segments are associated with the textual transcription of the speeches. For these purposes, a segmetation and annotation tools have been developed.

Most of the existing segmentation algorithms found in the specific literature are designed for general videos [8]. This means that they are complex algorithms prepared to detect the boundaries of the segment in all conditions. But in our context, a simple algorithm based on histogram comparison could work very well, as the case is. In this paper we briefly outline the algorithm itself, how it has been improved, as well as the main features of the segmentation and annotation tools.

Therefore, in this paper we describe an integrated system for searching the documents composing the digital library of the Parliament of Andalusia, composed of the PDF to XML converter, the video segmentation and annotation tools, and the search engine, as well as the way in which the user interacts with them. With this aim in mind, this paper is articulated as follows: the next section will introduce the general architecture of the integrated system. Section 3 explains the converter of PDF documents to XML. The segmentation algorithm as well as the annotation tool are described in Section 4. The search engine, and the model in which is based on are discussed in Section 5. Next, this paper outlines the search interface (Section 6), and the ends with some conclusions and future research lines (Section 7).

## 2 General Overview of the System Architecture

In order to offer a general overview of the elements that comprise the system, its general architecture is presented. Figure 1 shows a graphical representation of the system.

The Parliament publishes the official documents in PDF format. This is not the most appropriate format for structured retrieval. Then the first step is to transform each PDF file in a XML file, where the internal organization of these types of documents is captured and represented by means of XML tags, so all the content of the documents is structured. This format will allow the search engine to access the most appropriate units of textual information given a query.

With respect to the videos of the sessions, as the main objective is to give the user the possibility of accessing not only the most appropriate unit of text but also the piece of video associated to that text, all the units of the XML documents must be synchro-

nized with their corresponding portions of videos. To achieve this, a previous step is the division of the videos in segments. In the case of this regional chamber, as there are only four cameras recording the sessions, the realization of the recordings is really simple, so the segments will coincide with the changes of cameras. A segmentation module will be in charge of this task, giving as a result the segments and their keyframes (the automatically obtained segmentation could be edited manually). The next step is the synchronization of text and video. By means of an annotation tool, an expert user will proceed to visually associate each segment with the corresponding XML tag containing the transcription of the audio of the video segment. The output of this process is the XML of a session with time tags, indicating where the beginning and the end of the corresponding speech is located in the video.

In order to assure a fast downloading time and a direct access to the segements of the videos, we use the Flash format for them. Therefore, we have to apply another converter to transform the videos from their original AVI format into Flash format. This converter also adds time tags in the videos.

The search engine, which lies on Garnata [4], an Information Retrieval System for structured documents based on Bayesian Networks and Influence Diagrams [7], is in charge of retriving the relevant parts of the documents given a query. This piece of software has got a Web interface to interact with the user [1]. She/he formulates a query using a form, where the needed information could be described. The search engine takes this query and compute the relevance of all the elements contained in the XML collection. Finally, Garnata shows all the structural units in the Web page, sorted by decreasing value of relevance degrees. For each result, the text of the retrieved element, a link to the PDF document, a link to the XML file and if it has an associated video, a link to reproduce the corresponding portion of video are shown.

## 3 The PDF to XML Converter

In order to be able to perform structured retrieval, we have to convert the digital library of the Parliament in PDF format into XML. With this conversion, we are transforming the text contained in the PDF files, extracting the contents and placing them in the corresponding parts of the well defined structure of these official documents.

The conversion process has the following steps: Firstly, we transform the PDF documents into text files using an external tool called *pdftotext*. Afterthat the converter, developed in Java, takes the text of these files and uses a *lexical analyzer* and a *syntax analyzer* to process the text, extracting the tokens and generating the grammar to detect these tokens. Finally, to create the XML files, we use the DOM API, based on the building of a tree in memory whose nodes are XML tags. Therefore when a token is detected a new node or a group of nodes are created in the DOM tree, generating all the hierarchical structure of the XML format. Finally, the tree is converted into the corresponding XML file and it is validated.

[1] http://irutai.ugr.es/WebParlamento/index.php

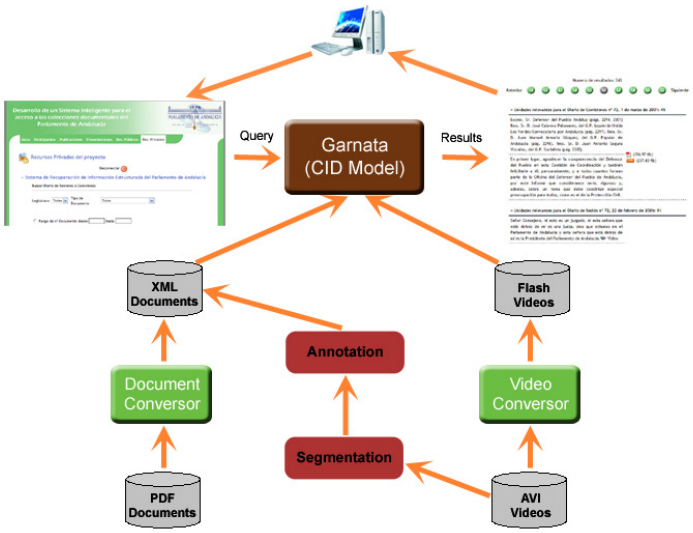


**Fig. 1.** System Architecture.

## 4 Video Segmentation and XML Annotation

Our aim is to find a method able to segment [6, 8] the videos of the Parliament of Andalusia, in a correct and efficient way, without forgetting the complexity of the implementation, which has to be low. These videos show long scenes, with few movements, and sudden changes. This means that the analysis of objects in the scene [9] is not going to help, because the speakers are usually static in the talk. In the chamber of the Parliament, there are four fixed cameras, focussing to the speaker, a general view of the chamber, and two centred in the seats of the deputies. Therefore, the realization of these videos is usually very static, and therefore very easy to detect the changes of cameras. Considering the segmentation using shot differences [10, 11], we shall notice that in the changes of camera, these differences are large, and inside the same segment, the differences are low, because the cameras are static and the movement is almost null. Therefore, we shall use this last method, considering the colour as the feature in which the method will be based on, basically because of the simplicity of the method and good results.

The implemented segmentation algorithm is based on detecting differences between shots. More specifically, these differences will be given by the different colours of the shots. We shall adopt a grey scale representation. From the RGB images of the video, we get that representation using a simple transformation: I(i,j) = (R(i,j) + G(i,j) + B(i,j))/3, where I is the matrix that represents the image in grey scale; R, G and B are the matrixes that represent the levels of red, green and blue of the image, respectively, and i and j are the indexes of a pixel. The histogram of the image represents the number of pixels that have a specific grey level on it. Considering that 8 bits are used to represent the intensity of 256 tones of the grey scale, we could represent the image by means of vectors of 256

elements. If we denote H the vector, and H(i) the number of occurrences of the grey level i in the image, then, in order to determine if there has been a change between two shots S1 and S2, with histograms H1 and H2, respectively, we could compute the difference of both vectors: $(H1 - H2)[i] =\mid H1[i] - H2[i] \mid$. Computing the difference for each grey level, and summing up all of them, we have a scalar value of the difference between both. If this value is greater than a certain threshold, then there is a change in the shots.

This is a really simple algorithm as a set of shots are considered included in the same segment if the difference between their histograms is low. But experimenting with the videos of parliamentary sessions, we realised that sometimes differences of histograms between shots are high even when there are no camera changes. Therefore, mistakes might be made. A solution is the application of a convolution filter, which makes that each element of the vector is the sum of those closest elements: $H1[i] = 0.1 * H1[i-2] + 0.2 * H1[i-1] + 0.4 * H1[i] + 0.2 * H1[i+1] + 0.1 * H1[i+2]$ . An important decision that will clearly have a great influence in the performance of the segmentation is the selection of the threshold value. It will depend on the resolution of the video, as in shots with a higher resolution the difference of their histograms will be proportionally larger. Moreover, images with a higher number of colours will also present a higher difference among shots, so we will have to consider the number of tones contained in the images. Therefore, the threshold used in our algorithm is defined as: $T = (Width * Height *$ No. of colours$)/K$, where $K$ is a parameter decisive to get an optimal segmentation.

For the type of considered videos, the value of the $K$ parameter has been obtained empirically studying several of them. The process which has led us to get it has been the following: first of all, we have obtained the difference for each pair of contiguous shots in each video; secondly, we have localized manually the cuts (changes of scene) that the segmentation algorithm should detect; and finally, once we have studied the values of differences in the shots in which there is a cut, we have selected a value for $K$ such as the threshold value is sufficiently low to detect all the real cuts, and sufficiently high to not detect cuts which does not exists. With a threshold of $16,000$ all the cuts are detected, and nothing except real cuts will be detected in the videos of the Parliament.

This basic segmentation algorithm works properly, but not efficiently. As the videos from the sessions of the Parliament of Andalusia are very long (about 5 hours), it is required to improve the segmentation speed, but without worsening the effectiveness. The first attempt is to reduce the number of shots to be considered. Instead of analysing each pair of them, we will discard $s$ shots between each studied pair. The next step will be to refine the segmentation to locate exactly where the cut is produced. This process is much faster than comparing each single pair of shots and also offers the optimal result.

A second optimization is related to the size of the image. If the difference of histograms with the full image is enough to differentiate shots, we could suppose that the histogram of only a portion of the image could offer enough information to perform this action. Then the reduction would improve the efficiency of the process, as the number of computations is lower. In the case of the videos of the Parliament of Andalusia, where the location of the cameras is known, we could know the part of the image that will suffer less interferences of movements. Then, if we divide the image in four quadrants,

the most appropriate section will be the lower left quadrant, using the pixels of this area to compute the histogram.

Once the automatic segmentation of a video has finished, the software offers the possibility of editing the segmentation manually. The output of this process is a set of segments, represented by a keyframe. The user may need to adjust the segmentation to prepare the posterior process of annotation, in order to be more accurate. Therefore, the user is allowed to edit the segments, combining them if they are contiguous, or dividing segments in two. In the application, all the segments found by the algorithm are shown in a window (Figure 2). More specifically, the keyframe of each segment. If we click in one of them, then all the shots contained in it are shown in a separate window. By means of submenus activated by the left button of the mouse, the user could edit the segments. There is also implemented a viewer that allows to play any segment.

When the posterior manual edition is over, the user is ready to carry out the annotation stage. The input of this process will be the sets of segments found in the video corresponding to a parliamentary session and the transcription of the speeches given in the chamber for that video. This transcription is represented by means of an XML document, which contains the structure of the session, as well as the text itself. The output will be the XML document containing the transcription synchronized with the video by means of time tags in the elements of the document. The segment of the video related to a specific text could be easily accessed. The annotation tool will consist of the manual association of segments with the corresponding elements in the XML document, so each tag will have a link to its corresponding part of the video.

**Fig. 2.** User interface for the segmentation tool.

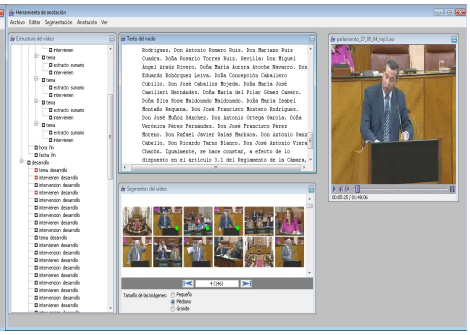

**Fig. 3.** User interface for the annotation tool.

Figure 3 shows the user interface of the annotation tool. It is composed of four windows. The left window shows the tree representation of an XML document containing a session diary. If a leaf node is clicked, then the text contained in it is shown in the central uppest window. The window below contains the segments found in the first part of the process. Finally, a player is included in the right part of the interface, in order to help the annotation. The annotation process is as follows: the user selects a segment in the video, then find the node in the XML document containing the transcription of the audio of that segment, and by means of a drag and drop action, associate the former with the latter. These steps are repeated until all the segments have been assigned a node

of the document. Actually, with the association of a segment to an XML element of the document, we introduce a pair of attributes to the corresponding tags, containing the beginning and ending times of the segment. This information will be enough to access the portion of the video in retrieval time. Once all the segments have been assigned leaf nodes of the XML tree, and therefore, all the affected tags have been complemented with temporal attributes linking the text with the video, it is necessary to propagate the times to upper nodes until reaching the root node.

## 5 The Search Engine: Garnata

The search engine to retrieve the relevant material for the user is Garnata [4], an Information Retrieval System, specially designed to work with structured documents in XML. This system is based on the Context-based Influence Diagram model (CID model) [3], which is supported by Influence Diagrams [5, 7]. These are probabilistic graphical models specially designed for decision problems.

An Influence Diagram (ID) provides a simple notation for creating decision models by clarifying the qualitative issues of the factors which need to be considered and how they are related, i.e. an intuitive representation of the model. It has also associated an underlying quantitative representation in order to measure the strength of the relationships. More formally, an influence diagram is an acyclic directed graph containing three types of nodes (decision, chance and utility) and two types of arcs (influence and informative arcs). The goal of influence diagram modeling is to choose the alternative decision that will lead to the highest expected gain (utility), i.e. the *optimal policy*. In order to compute the solution, for each sequence of decisions, the utilities of its uncertain consequences are weighted with the probabilities that these consequences will occur.

With respect to the CID model, starting from a document collection containing a set of documents, $\mathcal{D}$, and the set of terms, $\mathcal{T}$, used to index these documents, then we assume that each document is organized hierarchically, representing structural associations of its elements, which will be called *structural units*. Each structural unit is composed of other smaller structural units, except some 'terminal' or 'minimal' units which are indivisible, they do not contain any other unit, but they are composed of terms. Conversely, each structural unit, except the one corresponding to the complete document, is included in only one structural unit.

The chance nodes of the ID are the terms, $T_j$, and the structural units, $U_i$. They have associated a binary random variable, whose values could be term/unit is not relevant or is relevant, respectively.

Regarding the arcs, there is an arc from a given node (either term or structural unit) to the particular structural unit node it belongs to, expressing the fact that the relevance of a given structural unit to the user will depend on the relevance values of the different elements (units or terms) that comprise it.

Decision nodes, $R_i$, model the decision variables. There will be one node for each structural unit. It represents the decision variable related to whether or not to return the corresponding structural unit to the user, taking the values 'retrieve the unit' or 'do not retrieve the unit'. Finally, utility nodes, $V_i$. We shall also consider one utility node

for each structural unit, and it will measure the value of utility of the corresponding decision.

In addition to the arcs between chance nodes, we shall consider two different set of arcs. In order to represent that the utility function of a decision node obviously depends on the decision made and the relevance value of the structural unit considered, we use arcs from each chance node $U_i$ and decision node $R_i$ to the utility node $V_i$. Another important set of arcs are those going from the unit where $U_i$ is contained to $V_i$, which represent that the utility of the decision about retrieving the unit $U_i$ also depends on the relevance of the unit which contains it. Finally, for each node $V_i$, the associated utility functions must be defined. In Figure 4, an example of the topology of the CID model is shown.

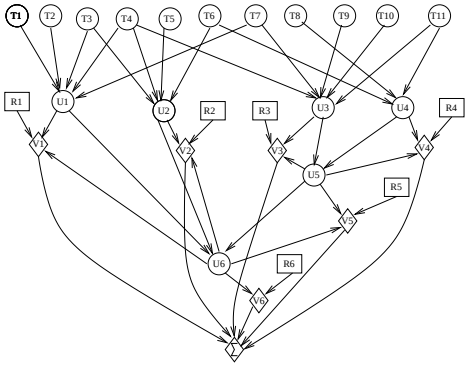


**Fig. 4.** An example of the CID model.

To solve an influence diagram, the expected utility of each possible decision has to be computed, thus making decisions which maximize the expected utility. In our case, the situation of interest corresponds with the information provided by the user when he/she formulates a query, $Q$, so we wish to compute the expected utility of each decision given the query. In the context of a typical decision making problem, once the expected utilities are computed, the decision with greatest utility is chosen: this would mean to retrieve the structural unit $U_i$ if the expected utility of retrieving is greater than the expected utility of not retrieving, and not to retrieve otherwise.

## 6 The Search Interface: Interacting with the User

The user interface of the search engine is based on a web page (http://irutai.ugr.es/-WebParlamento), where a user, who wants to get some information from the legislative collection of the Parliament of Andalusia, is able to express her/his information needs by means of a form (see Figure 5). The search parameters are the number of legislature, the kind of document (session diaries or official gazettes), publishing dates, range of documents, and finally, the query text. There is also the possibility of indicating how the results are arranged: a) Only one result for document: The system will show only

one result per document. This single document part should correspond to the best entry point for starting to read the relevant text in the document. b) All the results grouped by document: The search engine will return, for each document, all its relevant units sorted by their relevance degree. c) All the results: all the relevants units, without any association, presented to the user in decreasing order of their relevance degree.

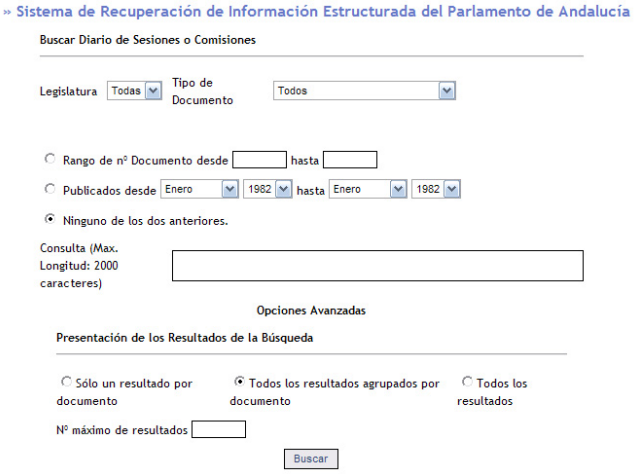


**Fig. 5.** User interface for searching.


**Fig. 6.** Results of a query.

Once, the search engine has computed the relevance degree of the structural units of the collection, the results are presented in a second web page in groups of ten. For each result, it is provided a brief portion of the text of the structural unit, a link to the corresponding PDF document that contains this unit, a link to the XML document displayed

in HTML format. Moreover, if a unit has an associated video, then there will be a link to this video, so the user will be able to watch the portion of the video corresponding to this structural unit. In Figure 6, we show an example of this presentation of results when it has been selected 'All the results grouped by document' option.

## 7 Conclusions and Further Research

This paper has presented an integrated software to access, from a structured retrieval perspective, the documents and videos generated by the Parliament of Andalusia, composed of all the tools needed to process these type of media: PDF to XML converter, video segmentation and annotation and search engine, as well as the graphical interfaces to interact with the user. We think the system has yielded good results until now but it is still in an experimental stage.

With respect to future works, we are planning the substitution of the video segmentation and annotation stages, by an automatic synchronization of audio and text. We are also working on the improvement of the retrieval capacity of the CID model.

## Acknowledgements

Work jointly supported by the Spanish Ministerio de Educación y Ciencia (TIN2005-02516), Consejería de Innovación, Ciencia y Empresa de la Junta de Andalucía (TIC-276), and Spanish research programme Consolider Ingenio 2010: MIPRCV (CSD2007-00018).

## References

1. R. Baeza-Yates, B. Ribeiro-Neto. Modern information Retrieval, Addison-Wesley. 1999.
2. Y. Chiaramella. Information retrieval and structured documents, Lectures on IR, Springer, 286-309. 2001.
3. L. M. de Campos, J. M. Fernández-Luna, J. F. Huete. Using context information in structured document retrieval: An approach using Influence Diagrams. IP&M. 40(5), 829 – 847, 2004.
4. L. de Campos, J. M. Fernández-Luna, J. Huete, and A. Romero. Garnata: An information retrieval system for structured documents based on probabilistic graphical models. In Proceedings of the IMPU'06 conference, 1024–1031, 2006.
5. F. V. Jensen. Bayesian Networks and Decision Graphs. Springer-Verlag, 2001.
6. I. Koprinska, S. Carrato. Temporal video segmentation: A survey. Signal Processing: Image Communication, 16(5), 477–500, 2001.
7. R. D. Shachter. Probabilistic inference and influence diagrams. Oper. Res., 36(4), 589–604, 1988.
8. F. Camastra, A. Vinciarelli. Video Segmentation and Keyframe Extraction. In Advanced Information and Knowledge Processing. Springer, 413–430, 2007.
9. Y. Lu, W. Gao, F. Wu. Automatic video segmentation using a novel background model. In Proc. of ISCAS. 2002.
10. A. Jain, S. Chaudhuri, A Fast Method for Textual Annotation of Compressed Video. In Proc. of ICVGIP. 2002.
11. A. Hanjalic, R. Lagendijk, J. Biemond. Automated Segmentation of Movies into Logical Story Units. Information Systems, 31(7), 638 – 658, 2006.