Workshop on Multimedia Discovery and Mining [MDM 2003]

at ECML/PKDD-2003

Edited by
Dunja Mladenić and Gerhard Paaß

ECML/PKDD 2003, Dubrovnik

ECML/PKDD 2003 Workshop on Multimedia Discovery and Mining [MDM 2003]

Preface

For Machine Learning and Knowledge Discovery, multimedia mining brings new challenges as data is extremely large in size, uses different modalities and has a rich internal structure. Required technologies include speech recognition, text mining, video and image analysis, information retrieval, and summarization.

The workshop on Multimedia Discovery and Mining aims at addressing usage of the Knowledge Discovery and Mining methodology to multimedia. It can be characterized as the extraction of useful patterns from multimedia data, in order to interpret the contents and make useful decisions.

Multimedia data typically has a complex structure and cannot readily be processed as a whole by the available data mining algorithms. Therefore we can say that Multimedia Mining involves two basic steps:

- Extraction of appropriate features from the data characterizing the aspects of interest.
- Selection of Data Mining methods to identify the desired information.

As experience grows the features will be refined to benefit from the algorithms at hand. At the same time new data mining procedures will emerge which are more adapted to the complex nature of Multimedia data.

Each of the papers of this workshop uses a specific combination of features and mining algorithms:

- Ceci, Berardi, and Malerba start with a symbolic description of a document layout and extract association rules between different layout characteristics.
- Szegő develops a logical methodology for describing and reasoning about the properties of XML documents containing multimedia contents.
- Saidi uses constraint satisfaction approaches to analyse training documents, e.g. typographically tagged print documents, and generates a set of rules that recognize similar documents.
- Grimaldi, Cunningham, and Kokaram characterize music by a variant of wavelets and apply different classifiers to find the genre of the music, e.g. "Classic".
- Leopold, Kindermann, Paaß, Vollmer, Cavet, Larson, Eickeler, and Kastner use speech recognition as well as color clusters to capture the contents of news videos and classify them into thematic categories using the support vector machine.

- Dorado and Izquierdo characterize video data by features like shot length and motion activity during shots and employ fuzzy rules to detect higher semantic concepts like "anchorperson".
- Jianfeng, Yang, and Zhanhuai describe Multimedia content in the hierarchical framework of the MPEG7 standard and show how complex retrieval operations may be implemented.

A related and very important effort to advance mining methods for multimedia is the annual TREC Video Retrieval Evaluation (http://www-nlpir.nist.gov/projects/trecvid/). It is devoted to research in automatic segmentation, indexing, and content-based retrieval of digital video. The invited talk for this workshop will be give by Alex Hauptman (CMU), one of the most successful participants of this evaluation.

We hope that this workshop will bring together researchers and practitioners interested in mining different type of multimedia data, e.g. users working in the media industry, experts on the technical background and standards of multimedia as well as data mining and machine learning specialists.

By Dunja Mladenić and Gerhard Paaß, July 2003

Organization

Program Chairs

- **Dunja Mladenić**, J.Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
- **Gerhard Paaß**, Fraunhofer Institute Autonomous Intelligent Systems, Schloß Birlinghoven 53754 St. Augustin, Germany

Program Committee

- **Marie-Aude Aufaure**, INRIA, Domaine de Voluceau, France
- **Alberto Del Bimbo**, Università degli Studi di Firenze, Italy
- **Joachim Diederich**, The University of Queensland, Australia
- **Chabane Djeraba**, University of Nantes, France,
- **Stefan Eickeler**, Fraunhofer Institute of Media Communication, Germany
- **Gareth Jones**, University of Exeter, UK
- **Thorsten Joachims**, Cornell University, USA
- **Odej Kao**, University of Paderborn, Germany
- **Gholamreza Nakhaeizadeh** , Daimler-Benz AG, Germany
- **Simeon Simoff**, University of Technology Sydney, Australia

Table of Contents

Alexander Hauptmann:
Machine Learning for Video Classification and Retrieval. p. 6

M. Ceci, M. Berardi, D. Malerba:
Mining Association Rules in Document Images. p. 7

D. Szegő:
A Logical Framework for Analyzing Properties of Multimedia Web Documents. p. 19

A. Saidi:
A Constraint Satisfaction Framework for Documents Recognition. p. 31

M. Grimaldi, P. Cunningham, A. Kokaram:
An Evaluation of Alternative Feature Selection Strategies and Ensemble Techniques for Classifying Music. p. 44

E. Leopold, J. Kindermann, G. Paaß, S. Vollmer, R. Cavet, M. Larson, S. Eickeler, T. Kastner:
Integrated Classification of Audio, Video and Speech using Partitions of Low-Level Features. p. 56

A. Dorado, E. Izquierdo:
Knowledge Representation of Low Level Features for Semantic Video p. 72

Y. Jianfeng, Z. Yang, L. Zhanhuai:
A Novel Method to Describe Multimedia Documents. Department of Computer Science & Engineering, p. 84

Machine Learning for Video Classification and Retrieval

Alexander Hauptmann
Dept. of Computer Science, Carnegie Mellon University,
Pittsburgh, PA 15213 USA

Video analysis and retrieval from video collection is a difficult task. One would like to characterize the video in terms of various types and styles, understand what objects are in the video, divide it into camera shots and group those into coherent scenes. Ultimately we want to make video as tractable and 'searchable' as current text collections that are indexed through the web. Video analysis is orders of magnitude more complex than speech recognition, where the data stream is merely a one-dimensional signal and suitable intermediate level representations such as sentences, words, and phonemes permit a divide-and-conquer approach utilizing machine learning.

However, there are also great opportunities in the multimedia analysis of video. Large amounts of data are available, although accurate annotations are sparse, a core set of feature extraction techniques are now well established, we can exploit simultaneous, correlated channels (audio & video) to extract more information and there is a large number of easy-to-use learning and mining software tools to make use of the annotated training material.

The pervasiveness of machine learning/data mining approaches can be illustrated in a range of tasks at different levels of video analysis, including:

- Learning of camera shot boundaries,
- Classifying video into common 'semantic' categories (e.g. outdoor, sports, news anchor person, …)
- Video retrieval as a classification problem through pseudo-relevance feedback

Ultimately, a user in an interactive retrieval situation will make use of the output of all these pieces of 'metadata'. The Informedia Digital Video Library system demonstrates the integration of multiple approaches and classes of features at different levels for the purpose of browsing through and retrieving from large broadcast video collections using image, text and audio information.

Some of the fundamental open problems in video analysis remain, such as dealing with features that are very low level (e.g. RGB pixel color values), a sometimes bewildering variety of machine learning approaches, each with different strengths and weaknesses as well as tuneable parameter settings, the combinations of information across multiple media (combining video and audio information into one comprehensive score), and the combination of different types of data in separate collections (e.g. documents from an OCR library and a video library need to be presented in a single ranked list).

In my talk I will illustrate these points with actual research results, as well as demonstrate the integration of all aspects into a complete system for video retrieval, which processes broadcast news on a daily basis, analyzes the recorded video in multiple ways, indexes all the extracted information into a comprehensive database. Finally a user is able to query, navigate, and summarize the accumulated data to find relevant and interesting information contained in the video.

Mining association rules in document images

Michelangelo Ceci Margherita Berardi Donato Malerba

Dipartimento di Informatica – Università degli Studi di Bari
via Orabona 4 - 70126 Bari
{ceci, berardi, malerba}@di.uniba.it

Abstract. In this paper we investigate the discovery of association rules from a particular kind of images, namely document images. Document images are initially processed to extract both their layout structures and their logical structures. To take into account the inherently spatial nature of the layout structure, a spatial data mining algorithm is applied, which returns spatial association rules. We illustrate and comment experimental results on a set of multi-page documents extracted by IEEE Transaction on Pattern Analysis and Machine Intelligence.

1. Introduction

The problem of mining association rules was originally introduced in the work by Agrawal et al. [1]. In its basic formulation, a database D of transactions is given, which is represented as a Boolean relational table. Each row in the table correspons to a transaction (a record); each columns corresponds to an item (or attribute); the i-th entry in a row contains true or false depending whether the item i is present in the corresponding transactions or not. An association rule is expressed by an implication:

$$X \rightarrow Y$$

where X and Y are a sets of *items*, such that $X \cap Y = \varnothing$. It means that whenever a transaction $T \in D$ contains X than T probably contains Y also. The conjunction $X \wedge Y$ is called *pattern*. In the support-confidence framework, an association rule is characterized by a pair of parameters, namely the *support*, which estimates the probability $p(X \subseteq T \wedge Y \subseteq T)$, and the *confidence*, which estimates the probability $p(Y \subseteq T \mid X \subseteq T)$. The problem of mining for association rules can be stated in the following way. Given two thresholds for support and confidence, *minsup* and *minconf*, enumerate all rules from D, whose support and confidence are, respectively, greater than *minsup* and *minconf*. A pattern $X \wedge Y$ whose support is greater than or equal to *minsup* is said to be *large* (or *frequent*). An association rule $X \rightarrow Y$ is *strong* if it has a large support (i.e. $X \wedge Y$ is large) and high confidence.

Traditionally, association rules are discovered for market basket analysis. However, it is becoming clear that they can be successfully applied to a wide range of domains, including for mining knowledge from images [13], where a transaction correspond to an image, and the items are objects recognized in the segmented image.

Mined association rules refer to the co-occurrence of objects in an image, and no spatial relationship between objects in the same image is considered.

In this paper we investigate the discovery of association rules from a particular kind of images, namely *document images*. Document images are raster images of paper documents acquired through scanning. Documents can be business letters, articles published on journals, invoices, and so on, and are characterized by the prevalence of textual information. This means that both text and graphics may be present in a document image, and that they both may convey relevant information for the application at hand. Nonetheless, the graphical components (images, lines, table frames, and so on) can be optional, while the textual components are not.

Recovering the symbol structure of digital images scanned from paper or produced by computer is the main goal of a research area known as Document Image Analysis (DIA). It is essentially considered as engineering discipline [12], because it systematically applies scientific knowledge to resolve conflicting constraints and requirements for problems of document image processing. From a document engineering perspective, it is important to create a disciplined approach to transform a paper document into some symbol structure (e.g., XML format). Altamura et al. [2] propose a decomposition of the data transformation process into six phases with a clear definition of both input and output.

First, the document images of a multi-page document have to be scanned, pre-processed and segmented to separate the background from the content. Then, a content-based classification of each extracted segment is required to separate text from graphics. A layout analysis process follows to extract the *layout* (or *geometrical*) *structure* of the whole document. The layout structure associates the content of a document with a hierarchy of layout components, such as blocks, lines, and paragraphs. It is related to the presentation of the document on some media. It is distinct from the *logical structure*, which associates the content of a document with a hierarchy of logical components, such as sender/receiver of a business letter, title/authors of a scientific article, and so on. It is related to the organization of the content. Luckily, in many documents the two structures are strongly related. This means that layout clues can be profitably used to reconstruct the logical structure of the document without “reading” the document itself. The general process of defining the mapping is called *document image understanding*[1] (or *interpretation*) [15], while the specific association of the whole document (root of the layout tree) with some class (root of the logical structure) is called *document image classification* [5]. Document image classification and understanding represent the fourth step in the transformation process. The fifth is the application of an OCR system only to those textual components of interest for the application domain. Once all necessary information has been extracted, data on the geometrical structure as well as on the corresponding logical labels and OCR’ed text can be stored in an XML document and properly rendered as the original document image by means of an automatically generated style sheet file (XSL and CSS) (sixth and last step).

[1] This process is distinct from *document understanding* which is related to natural language aspects of one-dimensional text flow.

All document processing steps reported above are supported by the DIA system WISDOM++ (http://www.di.uniba.it/_malerba/wisdom++/) [11]. This complex transformation process is knowledge-intensive and at the same time requires a high degree of adaptivity which can be achieved by means of machine learning techniques. Indeed, WISDOM++ embeds two machine-learning systems with different characteristics of expressive power and efficiency. The two systems supports the acquisition of the knowledge required in several processing steps.

In this paper we investigate an additional processing step, whose aim is that mining associations rules from document images. The discovery of association rules follows the processes of layout structure extraction (layout analysis) and logical structure extraction (document image understanding). We are interested in association rules expressing regularities among logical components of a set of document images belonging to the same class.

The extracted association rules can be used in a number of ways. First, new documents can be recognized as satisfying the constraints that define the domain template (document classification and retrieval). Indeed, recent approaches propose to use discovered association rules for classification tasks [8].

Second, the rules could be profitably used in the automatic layout correction. Currently, in WISDOM++ the automatic correction of the layout is performed by means of a set of rules learned from the sequence of user actions [3]. By formulating the problem as a planning problem, it is necessary to define both a goal and a metric evaluating the distance between the current state (layout structure) and the goal. This metric can be based on the number of association rules supported by the extracted layout structure.

Third, the rules could be also used in a generative way. For instance, if a part of the document is hidden or missing, strong spatial association rules can be used to predict the location of missing layout/logical components [7]. Moreover, a desirable property of a system that automatically generates textual documents is to take into account the layout specification during the generation process, since layout and wording generally interact [14]. Spatial association rules can be useful to define the layout specifications of such a system. Finally, this problem is also related to document reformatting [6].

The paper is organized as follows. In the next section the architecture of the DIA system WISDOM++ is briefly described. In Section 3 the approach followed to extract association rules is reported, while some experimental results are shown in Section 4.

2. Architecture of the system WISDOM++

The general architecture of WISDOM++, shown in Figure 1, integrates several components to perform all the steps reported in the previous section.

The *System Manager* manages the system by allowing user interaction and by coordinating the activity of all other components. It interfaces the system with the data base module in order to store intermediate information. The *System Manager* is

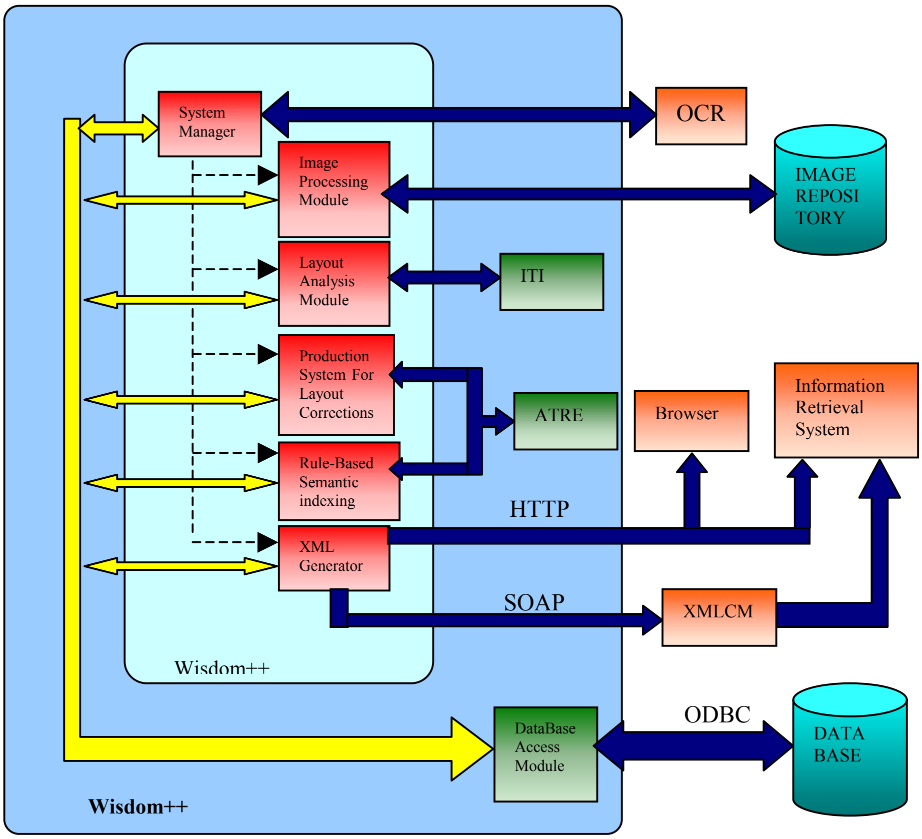


Fig. 1. Wisdom++ architecture.

also able to invoke the OCR on textual layout blocks which are relevant for the specific application (e.g., title or authors).

The *Image Processing Module* is in charge of the image preprocessing facilities. Preprocessing consists of a series of image-to-image transformations, which do not increase the system's knowledge of the contents of the document, but may help to extract it. One basic preprocessing step is the detection of the skew angle, which is defined as the orientation angle of the baselines of text blocks. Once the skew angle has been estimated the document image can be rotated to a reference direction to facilitate further format analysis and OCR. Additional preprocessing steps are noise filtering, such as removal of salt-and-pepper noise, and resolution reduction.

The *Layout Analysis Module* supports the separation of text from graphics and the layout analysis. The separation of text from graphics is performed into two steps: the segmentation detects non-overlapping rectangular blocks enclosing content portions, while the block classification identifies the content type (e.g., text, drawings, pictures and horizontal/vertical lines). The classification of blocks is based on the description of some features of each block. In WISDOM++ only geometrical (e.g., width, height, area, and eccentricity) and textural features are used to describe blocks. The classification of blocks as text, horizontal line, vertical line, picture (i.e., halftone images) and graphics (e.g., line drawings) is performed by means of the decision tree learning system ITI [16].

The *layout analysis* detects structures among blocks extracted during the segmentation step. It generates a hierarchy of abstract representations of the document image, the layout structure, which can be modeled by a *layout tree*. It is performed in two steps: firstly, the global analysis determines possible areas containing paragraphs, sections, columns, figures and tables, and secondly, the local analysis groups together blocks that possibly fall within the same area.

Once the layout analysis has been performed and the layout tree has been generated, the user can manually modify the layout tree by performing three different types of actions: vertical or horizontal split of a component in the layout tree, and grouping of two components. WISDOM++ stores both the result of corrective actions and the actions themselves. In this way it is possible to learn corrective layout operations from user interaction [3]. These operations are expressed as a set of "production" rules in the form of an antecedent and a consequent, where the antecedent expresses the precondition to the application of the rule and the consequent expresses the action to be performed in order to modify the layout structure. Production rules are then used by the *Production System for Layout Analysis Module*, which operates with a forward-chaining control structure. The production system is implemented with a theorem prover, using resolution to do forward chaining over a full first-order knowledge base. The system maintains a knowledge base (the working memory) of ground literals describing the layout tree. Ground literals are automatically generated by WISDOM++ after the execution of an operation. In each cycle, the system computes the subset of rules whose condition part is satisfied by the current contents of the working memory (*match phase*). Conflicts are solved by selecting the first rule in the subset.

The *Rule-based Semantic Indexing Module* performs the document classification and the document understanding tasks. By performing document image classification and understanding, WISDOM++ actually replaces the low-level image feature space (based on geometrical and textural features) with a higher-level semantic space. Query formulation can then be performed using these higher level semantics, which are much more comprehensible to the user than the low level image features [4]. Rules for document classification and understanding are learned by means of the inductive logic programming system ATRE [9], as explained in the next section.

The *XML Generator Module* is used to save the document in XML format. It transforms document images into XML format by integrating textual, graphical, layout and logical information extracted in the document analysis and understanding processes.

3. A spatial data mining approach

Differently from the work by Ordonez and Omiecinski [13], we also intend to take into account the inherent spatial nature of the layout structure, that is, we intend to discover, if any, spatial patterns between logical components. Therefore, association rule mining methods developed in the context of spatial data mining are considered [10].

There are three main peculiarities of the proposed approach. First, the spatial property of logical components is considered. Logical components are described in terms of their content type (e.g., text, graphics, etc.), their logical meaning (e.g., title, authors, etc.) and their geometrical features, which can be either relational or attibutional. Relational features relate two logical components on the basis of their mutual position in the document (e.g. *on_top(A,B), to_right(A,B)*). Attributional features refer to geometrical properties of layout components, such as width and height, as well as locational properties (position along x/y axis).

Second, the hierarchical structure of logical components is considered as well. It is possible to look at the set of logical components of a document image as a hierarchy where each single logical component could be related to another one by a *is_a* or a *part_of* relation (e.g *title* is part_of *identification*, *page_number* is_a *page_component*). The levels in the hierarchy are called granularity levels.

Third, the logical components can play different roles. Indeed, in spatial data mining attributes of some spatial objects in the neighborhood of, or contained in, a unit of analysis[2] may affect the values taken by attributes of the unit of analysis. Therefore, it is necessary to distinguish units of analysis, which are the *reference* objects of an analysis, from other *task-relevant* spatial objects, and it is important to represent interactions between them. In our application, some logical components play the role of *reference objects* while other logical components play the role of *task relevant objects*.

To mine spatial association rules we use SPADA (Spatial Pattern Discovery Algorithm) [10], which is based on a multi-relational data mining approach and permits the extraction of multi-level spatial association rules, that is, association rules involving spatial objects at different granularity levels. An example of association rule discovered by SPADA is:

```
is_a(A,running_head)→
                        on_top(A,B), is_a(B,content), type_text(A)
support: 90.9%   confidence: 90.9%
```

This rule means that if a logical component (*A*) is a *running_head* then it is textual and it is on top of another layout component (*B*) which is a component of type *content*. This rule has a high support and a high confidence (both expressed as percentages).

The problem of mining association rules by means of SPADA can be formally stated as follows:

Given

- a set of descriptions of the labelled documents (result of the document understanding)
- a set of reference objects *S*,
- some sets R_k, $1 \leq k \leq m$, of task-relevant objects,
- a background knowledge *BK* including some spatial hierarchies H_k on objects in R_k and a domain specific knowledge,
- *M* granularity levels in the descriptions (1 is the highest while M is the lowest),

[2] The unit of analysis is the basic entity or object about which generalizations are to be made based on an analysis and for which data are collected in the form of variables.

- a set of granularity assignments ψ_k which associate each object in H_k with a granularity level,
- a couple of thresholds *minsup*[*l*] and *minconf*[*l*] for each granularity level,
- a declarative bias *DB* that constrains the search space,

Find

strong multi-level spatial association rules.

The description of the algorithm is beyond the scope of this paper. Further details can be found in [10] where the problem of mining association rules by means of SPADA has been investigated in the context of georeferenced census data.

An example of labeled document image is shown in Fig 2. The set of descriptions of the labelled documents is expressed in the form of first-order logic conditions. The use of the first-order logic is necessary because the feature-vector representation, typically adopted in statistical approaches, cannot render the relational features.

Spatial features (relations and attributes) are used to describe the logical structure of a document image. In particular, we mention *locational* features such as the coordinates of the centroid of a logical component (*x_pos_center*, *y_pos_center*), *geometrical* features such as the dimensions of a logical component (*width*, *height*), and *topological* features such as relations between two components (*on_top*, *to_right*, *alignment*). We use the *aspatial* feature *type_of* that specifies the content type of a logical component (e.g. *image*, *text*, *horizontal line*). In addition there are other aspatial features, called *logical* features which define the label associated to the logical components. They are: *affiliation, page_number, figure,caption, index_term, running_head, author, title, abstract, formulae, subsection_title, section_title, biografy, references, paragraph*, *table, undefined*. In the following we present an example of document description on which runs SPADA:

```
class(h,tpami),
is_a(a,running_head), is_a(b,title),...
page(h,first),
part_of(h,a),part_of(h,b)=true,...
width(a,390),width(b,490),...
height(a,7),height(b,54),...
type_text(a),type_text(b),...
x_pos_centre(a,215),x_pos_centre(b,288),...
y_pos_centre(a,26),y_pos_centre(b,83),...
on_top(a,b),on_top(b,c),on_top(b,d),...
to_right(e,f),...
only_left_col(a,l),only_left_col(b,e),
only_right_col(b,e),only_middle_col(b,e), ...
```

where *h* represents the page and *a,...,g* represent the logical components of the page. It is noteworthy that the features *class* and *page* are used to describe the page and the relation *part_of* is used to express the membership of a component to a page. Numerical features are automatically discretized before inferring spatial association rules.

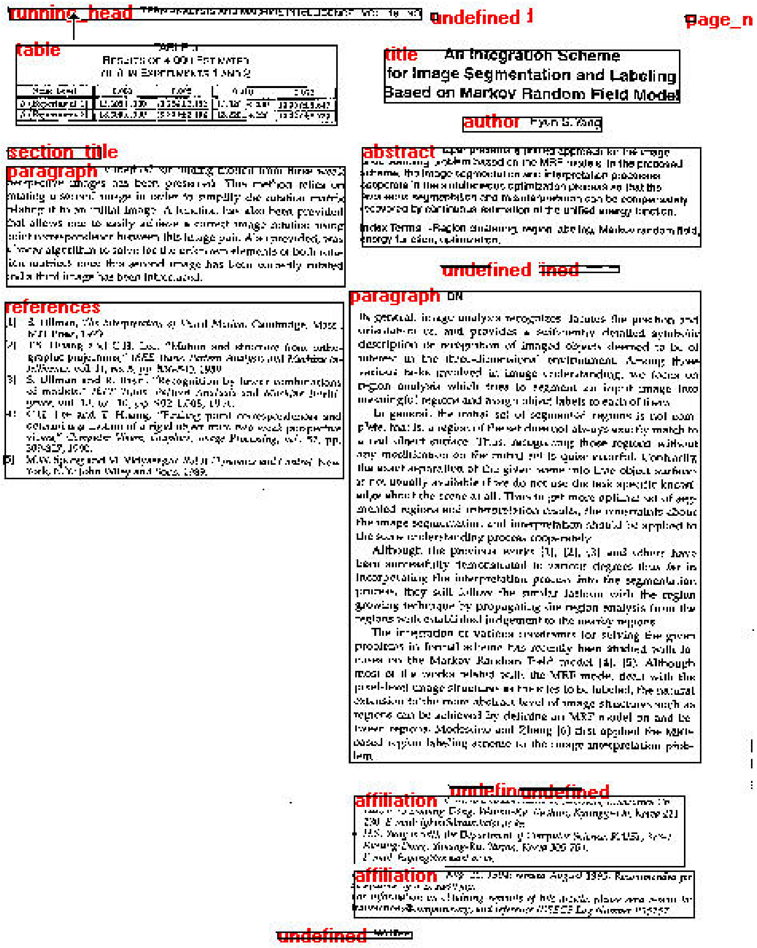


Fig. 2. An example of labeled document image.

The specification, by means of a set of Prolog rules, of the following domain specific knowledge:

```
at_page(X,first)          :- part_of(Y,X), page(Y,first)
at_page(X,intermediate)   :- part_of(Y,X), page(Y,intermediate)
at_page(X,last_but_one)   :- part_of(Y,X), page(Y,last_but_one)
at_page(X,last)           :- part_of(Y,X), page(Y,last)
```

permits to automatically associate information on page order to layout components, since the presence of some logical components may depend on the order page (e.g. *author* is in the first page). The specification of the hierarchy (Figure 3) allows the system to extract spatial association rules at different granularity size.

The declarative bias *DB* constrains the search space and aims at defining the *reference objects* (*ro*) and *task relevant objects* (*tro*). In our task, the *ro* are the logical

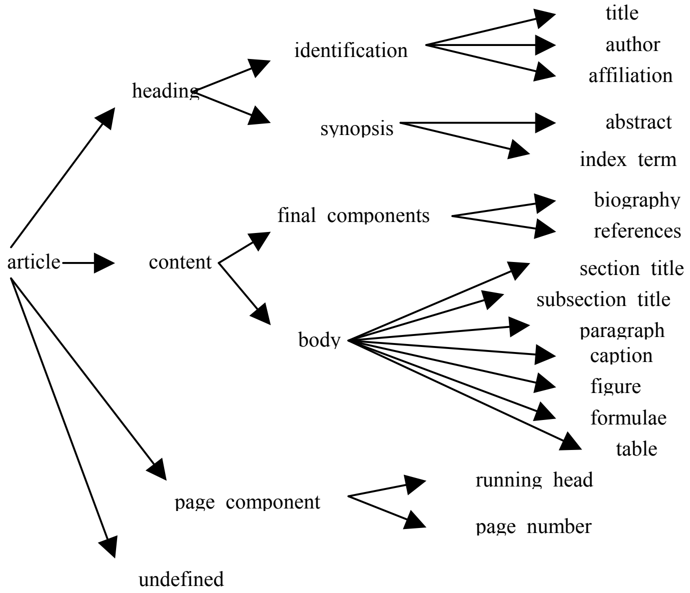


Fig. 3. Hierarchy of logical components.

components to which at least one satisfied logical feature, different from *undefined,* is associated. The *tro* are all the logical components.

4. Experimental results

Differently from [13], where the image mining system was evaluated on synthetic images automatically generated, we investigate the applicability of the proposed solution on real-world document images. In particular, we have considered six multi-page documents, which are scientific papers published as either regular or short in the IEEE Transactions on Pattern Analysis and Machine Intelligence in the January and February 1996 issues. Each paper is a multi-page document and has a variable number of pages and layout components for page. A user of WISDOM++ labels some layout components of this set of documents according to their logical meaning. Those layout components with no clear logical meaning are labelled as *undefined*. All logical labels belong to the lowest level of the hierarchy reported in the previous section. We processed 54 document images in all.

In Table 1 logical components distribution on the processed documents is shown. In particular, each table item reports the number of the logical components for the given document. The number of features to describe the six documents presented to SPADA is 17,880, about 331 features for each page document. The total number of logical components is 690 (114 of which are *undefined*) about 318 descriptors for each page document.

An example of association rule discovered by SPADA is:

Table 1. Labels distribution.

Document ID / Label	*1*	*3*	*4*	*6*	*7*	*9*	*Total*
affiliation	1	1	0	1	2	2	7
page_number	8	13	12	1	5	5	44
figure	19	13	12	3	12	12	71
caption	13	17	7	3	11	5	56
index_term	1	1	1	0	0	0	3
running_head	14	15	14	1	6	5	55
author	1	1	2	1	1	1	7
title	1	1	1	1	1	1	6
abstract	1	1	1	1	1	1	6
formulae	24	19	21	0	4	5	73
subsection_title	3	1	1	0	0	0	5
section_title	4	4	2	0	1	1	12
biografy	2	1	1	0	0	0	4
references	3	3	2	1	1	2	12
paragraph	54	55	50	3	19	21	202
table	0	9	1	0	2	1	13
undefined	21	26	27	10	14	16	114

```
is_a(A,author) → only_middle_col(A,B) ,
             is_a(B,heading), height(B,[1..174]), type_text(A)
support: 85.71% confidence: 85.71%
```

The spatial pattern of this rule involves six out of seven (i.e. 85.71%) blocks labelled as authors. This means that six logical components which represent the *author* of some paper are textual components vertically centered with a logical component B at the *heading* of the paper, with height between 1 and 174.

At a lower granularity level, a similar rule is found where the logical component *B* is specialized as *abstract*:

```
is_a(A,author) → only_middle_col (A,B),
            is_a(B,abstract), height(B,[1..174]), type_text(A)
support: 85.71%  confidence: 85.71%
```

The rule has the same confidence and support reported for the rule inferred at the first granularity level.

Another example of association rule is:

```
  is_a(A,index_term) → on_top(A,B), is_a(B,content),
                                              height(B,[1..174])
support: 100.0 confidence: 100.0
```

which states that all logical components *index term* are above a logical component of type content (body of the paper or final components). At the lowest granularity level the following rule is found:

```
is_a(A,index_term) →  on_top(A,B),
                    is_a(B,section_title), height(B,[1..174])
support: 100.0 confidence: 100.0
```

It discloses the fact that the specific fact that the *content* component associated to an *index_term* component in the previous rule is a *section_title* component.

Table 2. Number of rules.

No of Rules	Level 1	Level 2	Level 3	Level 4
min_conf	*0.5*	*0.5*	*0.5*	*0.5*
min_supp	*0.8*	*0.8*	*0.8*	*0.8*
Affiliation	8	8	8	8
Page_Number	16	16	16	16
Figure	10	10	10	5
Caption	7	7	7	1
Index_term	34	34	34	34
Running_head	11	10	9	2
Author	38	39	39	39
Title	55	55	56	52
Abstract	46	69	69	69
Formulae	15	15	15	5
Subsection_title	41	41	41	41
Section_Title	18	18	14	12
Biografy	8	8	8	8
Paragraph	24	15	15	4
Table	13	13	13	4
TOTAL	**344**	**358**	**354**	**300**

The high support and confidence of both rules is mainly due to the very low number of index terms in the documents selected for the experimentation.

A final example of association rule discovered for paragraphs is the following:

```
is_a(A,paragraph), on_top(A,B) →
                is_a(B,body), width(A,[256..383]) type_text(A)
support: 53.73 confidence: 100.0
```

It states that 53,73% of textual paragraphs, whose width is between 256 and 383 pixels, are above a body component.

The number of mined association rules for each logical component at different granularity levels is reported in Table 2. Although the thresholds for the minimum support and the minimum confidence are quite high (0.8 and 0.5, respectively, for all levels) SPADA has found several spatial associations involving all logical components, references excluded. Many spatial patterns involving logical components (e.g., affiliation, title, author, abstract and index term) in the first page of an article are found. This can be explained by the observation that the first page generally has a more regular layout structure and contains several distinct logical components.

5. Conclusions

This work presents an application of spatial data mining techniques to the problem of finding associations between logical components extracted from document images by means of document analysis and understanding methods. As future work, we intend to investigate the application of mined association rules in three different contexts, namely document classification and retrieval, automated layout correction, and automated generation of documents.

Acknowledgements

This work fulfills the research objectives set by the IST-1999-20882 project COLLATE (Collaboratory for Annotation, Indexing and Retrieval of Digitized Historical Archive Material) funded by the European Union.

References

1. Agrawal, R., Imielinski, T., & Swami, A.: Mining association rules between sets of items in large databases.*Proc. ACM-SIGMOD Conference*, Washington, DC, (1993).
2. Altamura O., Esposito F., & Malerba D.: Transforming paper documents into XML format with WISDOM++, *Int. Journal on Document Analysis and Recognition*, 4(1), 2-17, 2001.
3. Berardi, M., Ceci, M., Esposito, F., & Malerba, D.: Learning Logic Programs for Layout Analysis Correction. *Proceedings of the Twentieth International Conference on Machine Learning*, Washington, DC, (2003), in press.
4. Bradshaw, B.: Semantic based image retrieval: a probabilistic approach. *ACM Multimedia* 2000, (2000), 167-176.
5. Esposito F., Malerba D., Semeraro G., Annese E., and Scafuro G.: An experimental page layout recognition system for office document automatic classitication: An integrated approach for inductive generalization. *In Proc. of the Tenth Int. Conf on Pattern Recognition*, (1990), 557-562.
6. Hardman L., Rutledge L., & Bulterman D.: Automated generation of hypermedia presentation from pre-existing tagged media objects, *Proc. of the Second Workshop on Adaptive Hypertext and Hypermedia* (1998).
7. Hiraki, K., Gennari, J.H., Yamamoto, Y., and Anzai, Y., Learning Spatial Relations from Images, *Machine Learning Workshop,* Chicago, (1991), 407-411.
8. Liu, B., Hsu, W., & Ma, Y.: Integrating classification and association rule mining. *In KDD'98*, New York, NY (1998).
9. Malerba D., Esposito F., and Lisi F.A.: Learning recursive theories with ATRE, in H. Prade (Ed.), *Proceedings of the 13th European Conference on Artificial Intelligence*, John Wiley & Sons, Chichester, England, (1998), 435-439.
10. Malerba, D., Esposito, F., Lisi, F.A., & Appice, A.: Mining Spatial Association Rules in Census Data. *Research in Official Statistics*, 5(1), 19-44, 2002.
11. Malerba, D., Ceci, M., & Berardi, M.: XML and Knowledge Technologies for Semantic-Based Indexing of Paper Documents. In Marik, Retschitzegger, Stepankova (Eds.), *Proceedings of DEXA*, Springer, Berlin, (2003), in press.
12. Nagy, G.: Twenty Years of Document Image Analysis in PAMI. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(1), 38-62, 2000.
13. Ordonez, C., & Omiecinski, E.: Discovering association rules based on image content, *Proceedings of the IEEE Advances in Digital Libraries Conference 99* (1999).
14. Reichenberger, K., Rondhuis, K.J., Kleinz, J., & Bateman, J.: Effective Presentation of Information Through Page Layout: a Linguistically-Based Approach. *Proceedings of the ACM Workshop on Effective Abstractions in Multimedia.* San Francisco, CA, (1995).
15. Tsujimoto, S., & Asada, H.: Understanding Multi-articled Documents, *in Proc. of the Tenth Int. Conf. on Pattern Recognition*, Atlantic City, N.J., (1990), 551-556.
16. Utgoff, P.E.: An improved algorithm for incremental induction of decision trees. *Proc. of the Eleventh Int. Conf. on Machine Learning*, San Francisco,CA: Morgan Kaufmann (1994)

A Logical Framework for Analyzing Properties of Multimedia Web Documents

Dániel Szegő

Budapest University of Technology and Economics
Department of Measurement and Information Systems
H-1521, pf. 91, Budapest, Hungary
szegod@mit.bme.hu

Abstract. This paper describes some practical and theoretical foundations of Transformational Structured Document Logic (TSDL), which is a logical methodology for analyzing properties of Web documents, XML or HTML, consisting multimedia data, like image, natural language text, video or audio. TSDL can make benefits in searching, or in defining filters for multimedia Web documents. Both syntax and semantics of TSDL are described, and an efficient evaluation algorithm is also briefly introduced.

1 Introduction

During the last ten years, the success of World Wide Web was increasing and it has become part of our daily life. At the golden ages of WWW, pages mainly contained textual elements in well defined formats, like HTML or XML. In our days, this situation has become more complicated. Most Web documents contain non-textual elements either like images, videos, sounds or script language elements. These documents are usually referred as multimedia Web documents. Analyzing such a document is much more challenging than handling simple ones, because it requires knowledge of handling simultaneously different media and the structure of documents. For example, medical or geographical documents usually contain high-resolution images. Automatic processing of such a document surely requires the cooperation of image processing and traditional Web document techniques. Similarly, ornithological documents might include audio elements, which might be analyzed by signal processing techniques.

Transformational Structured Document Logic (TSDL) is a logical framework for analyzing properties of Web documents consisting of non-standard elements like image or audio components. TSDL itself does not realize signal processing, image processing or natural language understanding elements, but it provides a general logical framework in which these elements can easily be integrated. Hence, these elements can be cooperated with each other and with the structural part of a Web document to identify the existence of properties regarding to both structure and multimedia.

The property of a document simply means a true or false value which is true for some multimedia documents and false for the others. This property might seem to be a simple service; however, its importance cannot be overestimated. First of all, properties can be used in a searching process. A searching query can be implemented as a property, so documents for which the property is true provide the result of the query. This mechanism can be used in a simple WWW search, and it might be the basic query process of an XML database [1]. Secondly, properties of documents can be used to implement Web filters[1] [2,3,4]. The main purpose of a Web filter is to select a few pieces of information from the WWW and to present it to the user in a specific way (e.g. by WAP). As a simple example, we can imagine a businessman desiring monitor the money market. Properties can be used in both finding the necessary information and maintaining the information even if the source documents are reconstructed. Thirdly, documents can be validated with the help of properties, similarly as DTD (Document Type Definition) does for XML documents [5]. If a property is true for a given document, it represents a valid one, whilst non-valid documents are indicated by false properties. Unfortunately, existing validation techniques do not pay attention on multimedia elements [5]. Last but not least, automatic categorization or data mining of documents can be supported by property analysis [6]. For example, a simple categorization process can be realized by identifying the existence of a set of properties over a set of Web documents. Documents with given properties could form a category.

Several approaches were developed to analyze properties of Web documents. Some of them focus on the structure of documents [7,8,9], whilst others deal with the connection of several different but linked document [10,11]. However, they usually do not pay attention on multimedia elements. These elements are usually treated in the same way as other parts of the documents; no special multimedia processing appears.

The reminder of this paper is organized as follows. In section 2, basic concepts behind the logic and the basic architecture are demonstrated. Section 3 introduces mathematical foundations of the logic containing model, syntax, semantics and some demonstrating examples. An efficient algorithm for evaluating TSDL expressions is also proposed in section 3. Finally, section 4 draws some conclusions.

2 Basic Architecture

The basic architecture of TSDL mainly focuses on HTML and XML documents, however theoretically other formats could also be considered. Documents are read and parsed by XML and HTML parsers. Parsing produces a document model which can be considered as the inside representation of the analyzed document. This representation is a directed tree which nodes are tags of the document and edges represent the embedding of tags. Nodes of the graph are usually marked by attributes (which will also be called as atomic predicates in the followings). Multimedia elements are also represented as tags, but they might refer to other resources like files of a directory system or objects of a multimedia database. For example, images of an HTML document are usually stored as standalone jpeg files. Parsers produce only an initial document model which might be further modified by different kinds of transformations.

[1] TSDL was primarily motivated by the IKTA-0186 project which focused on studying and implementing Web filters.

Each transformation creates a new piece of document model which also must be a directed tree. Consequently, transformations can add and delete nodes from the document model or they can change attributes of nodes. Basic concepts of the architecture can be seen on Figure 1.

There are three major kinds of transformations.

1. *Pre-transformations* try to increase the quality of inside representation by adding further important features. As an example, most HTML documents do not follow the HTML specification, therefore transformations need to be used to get the necessary document model.
2. *Filters* delete unnecessary parts of document models. For example, if we are not interested in handling comments or script language nodes, an adequate filter can easily eliminate them.
3. Perhaps the most important transformation is the *multimedia analysis*. It may include transformations for images, sound or natural language texts. These transformations analyze the necessary subparts of the document model and create new models which contain information about the result of the analysis. For example, an image analysis may indicate that a picture is colored by adding an attribute to a node of the picture. Similarly, a natural language transformation can indicate topics of all natural language texts by adding some nodes and attributes to the original graph.

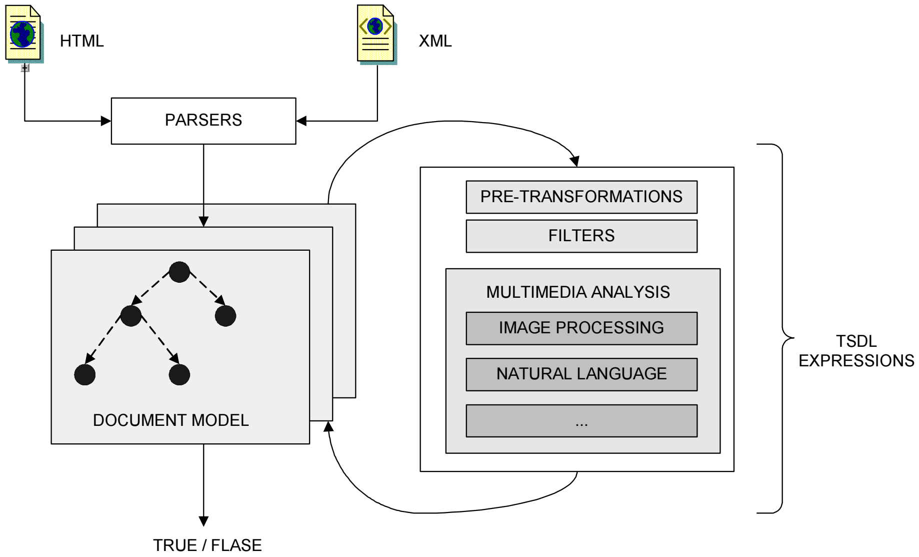


Fig. 1. Basic Architecture

Initially, there is only one document model, but different new models can also be created by applying different transformations. The exact order and number of transformations are controlled by TSDL expressions. Properties of multimedia Web documents are computed simultaneously from the initial and the newly created models.

3 Mathematical Foundations

This section introduces mathematical foundations of TSDL, which are based on the concepts of the previous architecture. The architecture can be formalized on two different levels. Firstly, mathematical model of Web documents can be expressed, which is called as *document level*. It describes the structure of individual documents, focusing mainly on tags and relations between tags. Secondly, transformation of documents, the *transformational level*, will be described, which analyzes interoperations of documents and transformations. Our approach is based on modal logic. Consequently, three main elements of the logic have to be analyzed: the model theory, the syntax and the semantics. All three elements will be described on both document and transformational level. From clear theoretical point of view document level can be regarded as the object level of the formalism, and transformations represent meta level description.

3.1 TSDL Model

The document level of the model is a simple directed tree-graph, which nodes are labeled by atomic predicates. Transformational model is an edge labeled directed graph. Nodes of the transformational level graph are document models and labels of edges are atomic transformations. If two document models are connected by an edge labeled by 't', it means that applying 't' transformation on the first document results in the second document. Theoretically, transformational level graph could be infinite; however, only finite subparts are taken into account at real problems.

The ***document model*** is a six tuple <V, AP, top, *p*, *c*, *ap*>.

- V is a set of nodes of the graph.
- AP is a set of atomic predicate, top$\in$V is the top node.
- p:V$\rightarrow$V is a partial map associating each node with its parent node.
- c:V$\rightarrow 2^V$ is a partial map associating each node with its set of children nodes.
- ap:V$\rightarrow 2^{AP}$ is a partial map associating each node with a set of atomic predicates.
- Paths of the graph are represented by $<v_1,v_2,v_3,\ldots v_{N-1},v_N>$ sequences, where $v_i \in V$, $p(v_i)=v_{i-1}$, and $v_{i+1} \in c(v_i)$.
- Each path of the graph must be circle free, each maximal long path has to start from the *t* top node, and each node of the graph must be reached from the top node through one of the paths.

This definition seems trivial for an XML document [5]. For example, tags can be translated to nodes and embedding of tags represents the parent-children mapping. This transformations is less trivial for an HTML document [12], consequently pre-transformations need to be applied. A multimedia element of a document initially appears as a node whose atomic predicates represent type and resource information.

Applying multimedia transformations, this node might get further atomic predicates or other related multimedia nodes might be added to the model. For example, an image element of an HTML document is a simple node with an ‘image’ and a file location atomic predicate. After applying a ‘resolution identification’ transformation, a new atomic predicate may be introduces indicating the resolution of the image (e.g. ‘high resolution’ or ‘low resolution’).

The ***transformational model*** is a graph whose nodes are document models, and the edges of the graph are labeled by atomic transformations. More formally, the transformational model is a three tuple $<D, T, \eta>$.

- D is a set of document models.
- T is a set of atomic transformations.
- η: $D\times D\rightarrow T$ is a partial map associating each pair of document models (edges) with an atomic transformation.
- Each transformation has to be deterministic: If $\eta(<d_1,d_2>)=\eta(<d_1, d_3>)$ then $d_2=d_3$ (where $d_1,d_2,d_3\in D$).
- Primary consequence of deterministic behavior is that the following notations can be used: $t_k(d_i) = d_j$, if and only if $\eta(<d_i,d_j>)=t_k(d_i,d_j\in D, t_k\in D)$.

Example 1. As a simple example, one can imagine a company whose confidential Web documents are marked by special confidentiality notes. There are two kinds of notes, an image and a natural language text, indicating that the given document is confidential. Although these notes represent the same content, it is not sure that they are equal bit by bit. For example, the confidentiality image might appear with different resolution, size or colors. Assume that we would like to develop a tool which identifies secret documents. In this case, at least three different kinds of media must be handled. Images must be analyzed by image processing, natural language texts by text analysis and the structure of the document by structure analysis. Analyzing such documents requires at least three different atomic transformations (see Figure 2.). Pre-transformation would eliminate all parts of the document which are irrelevant to confidentiality. Natural language analysis would identify the text and image processing the image of confidentiality.

The formal description of transformational and document model is the following:

- $\{d_0,d_1,d_2,d_3\}\subseteq D$, $T=\{ t_1,t_2,t_3\}$, $\{<d_0,d_1,t_1>,<d_1,d_2,t_2>,<d_1,d_3,t_3>\}\subseteq\eta$
- $d_1=<V_1, AP_1, top_1, p_1, c_1, ap_1>$, $V_1=\{v_1, v_2, v_3, v_4, v_5\}$,
  AP_1 ={‘top’,’table’,’emphasis’,’image’,’text’,’resource_ref1’,’resource_ref2’}
  $top_1=v_1$, $p_1=\{<v_2,v_1>,<v_3,v_1>,<v_4,v_2>,<v_5,v_3>\}$,
  $c_1=\{<v_1,\{v_2,v_3\}>,<v_2,\{v_4\}>,<v_3,\{v_5\}>\}$, ap_1={<v_1,{‘top’}>,
  <v_2,{ ‘table’}>,<v_3,{‘emphasis’}>,<v_4,{‘image’,’resource_ref1’},
  <v_5,{‘text’, ’resource_ref2’}>}
- $d_2=<V_2, AP_2, top_2, p_2, c_2, ap_2>$, $V_2=\{w_1, w_2, w_3, w_4, w_5\}$
  AP_2={‘top’,’table’,’emphasis’,’image’,’text’,’resource_ref1’,’resource_ref2’,’conf_text’ } $top_2=w_1$, $p_2=\{<w_2,w_1>,<w_3,w_1>,<w_4,w_2>,<w_5,w_3>\}$,
  $c_2=\{<w_1,\{w_2,w_3\}>,<w_2,\{w_4\}>,<w_3,\{w_5\}>\}$, ap_2={<w_1,{‘top’}>,<w_2,{‘table’}>,
  <w_3,{ ‘emphasis’}>,<w_4,{‘image’,’resource_ref1’},
  <w_5,{‘text’, ’resource_ref2’, ’conf_text’}>}

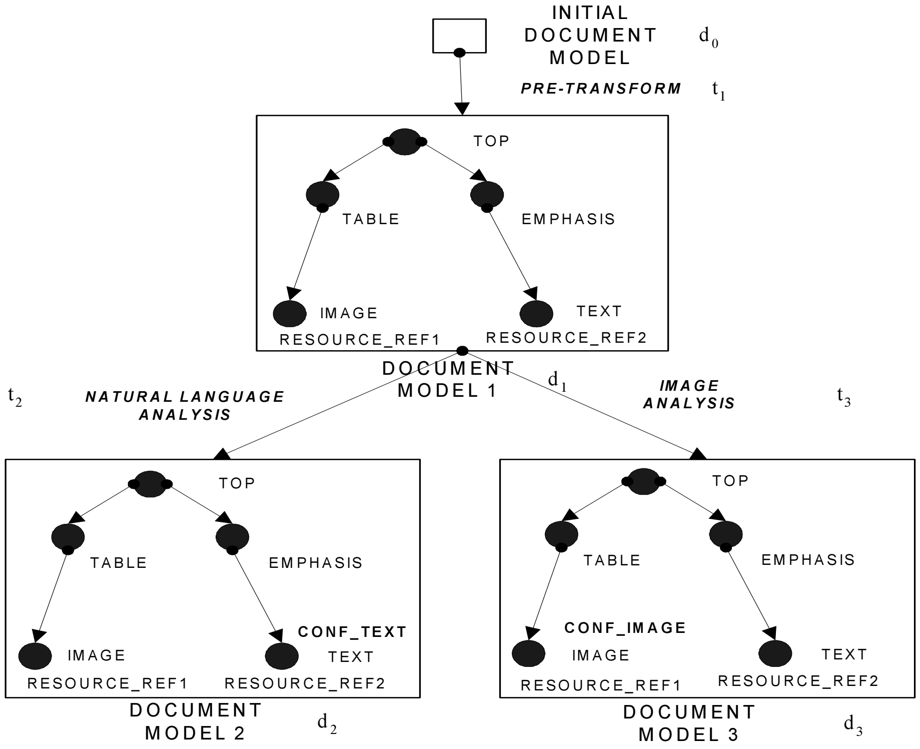


Fig. 2. Simple Example

- d_3=<V_3, AP_3, top_3, p_3, c_3, ap_3>, V_1={n_1, n_2, n_3, n_4, n_5}, AP_3={'top','table','emphasis','image','text','resource_ref1','resource_ref2', 'conf_image' } top_3=n_1, p_3={<n_2,n_1>,<n_3,n_1>,<n_4,n_2>,<n_5,n_3>}, c_3={<n_1,{n_2,n_3}>,<n_2,{n_4}>,<n_3,{n_5}>}, ap_3={<n_1,{'top'}>,<n_2,{ 'table'}>, <n_3,{ 'emphasis'}>,<n_4,{'image','resource_ref1','conf _image'}, <n_5,{'text', 'resource_ref2'}>}

It is important to note that transformational model is only partially determined in this simple example. For example, the structure of the initial document or the structure of the d_2 document after the application of one of the transformations is not known exactly. However, for solving real-world problems, this partial knowledge is sufficient.

3.2 Syntax and Semantics

Similarly to model theory, syntax and semantics are described at both document and transformational level. Document level focuses on expressing statements on standalone documents, including expressions on atomic predicates, nodes and relation of nodes. Transformational level focuses on statements of transformations and documents. The exact meaning of the ***syntax*** can be given with the semantics.

- CT ::= *{***t***}* | CT **;** *{***t***}*
- D :: = *{***a***}*| **T** | $\perp$ | D $\wedge$ D | D $\vee$ D | $\neg$ D | D **P** D | D $\mathbf{C}_\exists$ D | D $\mathbf{C}_\forall$ D
- TSDL :: = D | TSDL $\wedge$ TSDL | TSDL $\vee$ TSDL | <<CT>> TSDL

D describes the document level language and TSDL describes the transformational one. The '*{*a*}*' expression does not represent a syntactic form, but the abbreviation of one piece of atomic predicate. Similarly, '*{*t*}*' denotes one piece of atomic transformation. CT represents the language of complex transformations which is simply sequence of atomic transformations separated by ';'. T and $\perp$ represent the top and bottom of a document model which can be regarded as being true or false for the whole document. $\wedge$, $\vee$, $\neg$ are the basic logical operators. P should be read as the parent operator, $C_\exists$ is the exist-children and $C_\forall$ is the all-children operator. <<T>> is a transformational level expression called as deterministic execution of transformations.

The ***semantics of a document level expression*** is interpreted with an 'x' node of an 'M' document model. In a theoretical point of view, nodes of the document model represent possible worlds of the modal logic, parent and children maps realizes relations between possible worlds [13]. We can say that an 'x' node of a given document model 'M' satisfies an expression 'exp', denoted by M, x |= exp. In other words, 'exp' expression is true for 'x' node of 'M' model. An expression is true for an 'M' model if there is an 'x' node for which M, x |= exp.

- M, x |= T, for all $x \in V$ (where V is the node set of document model).
- M, x |= $\perp$, for none of the $x \in V$ nodes.
- M, x |= a, if and only if, $a \in ap(x)$.
- M, x |= $\neg$D, if and only if, not M, x |= D.
- M, x |= $D_1 \wedge D_2$, if and only if, M, x |= D_1 and M, x |= D_2.
- M, x |= $D_1 \vee D_2$, if and only if, M, x |= D_1 or M, x |= D_2.
- M, x |= D_1P D_2, if and only if, M, x |= D_1 and M, p(x) |= D_2.
- M, x |= $D_1 C_\exists D_2$, if and only if, M, x |= D_1 and exists an $y \in c(x)$ for which M, y |= D_2.
- M, x |= $D_1 C_\forall D_2$, if and only if, M, x |= D_1 and exists an $y \in c(x)$, and for all $y \in c(x)$ M, y |= D_2.

The ***semantics of a TSDL expression*** is interpreted with a 'd' document model of a 'TM' transformational model. We can say that a 'd' document model of a given transformational model 'TM' satisfies an expression 'exp', denoted by TM, x |= exp. In other words, 'exp' expression is true for a 'd' document model of 'TM' transformational model.

- TM, d |= exp, if exp is a D expressions and there is an 'x' node of 'd' document model for which d, x |= exp.
- TM, d |= $TSDL_1 \wedge TSDL_2$, if and only if, TM, d |= $TSDL_1$ and TM, d |= $TSDL_2$.
- TM, d |= $TSDL_1 \vee TSDL_2$, if and only if, TM, d |= $TSDL_1$ or TM, d |= $TSDL_2$.
- TM, d |= <<t_1;t_2;t_3;..;t_N>> TSDL, if and only if, there is a <d_0,d_1,d_2,…, d_N> sequence of document level models for which d=d_0, $\eta(d_{i-1},d_i) = t_i$ (for all $I \in \{1..N\}$) and TM, d_N |= TSDL.

In practical applications, there is an initial document model which is the starting node of the transformational level graph. In our case, this starting document is the direct output of HTML or XML parsers. It consists of all tags of the document as nodes and all texts or attributes as atomic predicates. TSDL expressions are usually evaluated over the initial document. Certainly, evaluating a TSDL expression requires transforming the existing documents so new document models will be created during the evaluation process. From a clear theoretical point of view, model theory of the logic is sufficient for evaluating properties over Web documents. Properties are formalized as TSDL expressions and evaluating the expressions clearly identifies whether a property holds or not. Therefore, proof theory of TSDL is not studied in this article.

Example 2. Considering the previous example (see Figure 2.), let us suppose that confidentiality text and image cannot be placed anywhere in the document but they must be highlighted somehow. For example, confidentiality image must be in a table and confidentiality text must be emphasized (for example by <em> or <strong> HTML tags). Assume that one would like to develop a document checking tool which filters out the documents with non-highlighted confidentiality notes (non-valid documents). It can also be imagined as a document categorization process where there are three categories: documents with no privacy notes, documents with non-valid privacy notes and documents with valid privacy notes. Realizing such a categorization requires the cooperation of different media analysis like image or natural language text processing with structural analysis of the document.
Different properties of Web documents, containing confidentiality text or image, can be easily analyzed by TSDL expressions:

- <<t_1;t_2>>conf_text expression is true for those documents which contain confidentiality text.
- <<t_1;t_3>>conf_image expression is true for those documents which contain confidentiality image.
- <<t_1>>((<<t_2>>conf_text)∨(<<t_3>>conf_image)) expression is true for those documents which contain confidentiality text or image.
- <<t_1;t_2>>(conf_ text P emphasis) expression is true for those documents which contain confidentiality text and this text is emphasized (valid confidentiality text).
- <<t_1;t_3>>(conf_image P table) expression is true for those documents which contain confidentiality image and this image is in a table (valid confidentiality image).
- <<t_1;t_2>>(conf_text P ¬emphasis) expression is true for those documents which contain confidentiality text but this text is not emphasized (non-valid confidentiality text).
- <<t_1;t_3>>(conf_image P ¬table) expression is true for those documents which contain confidentiality image and this image is not in a table (non-valid confidentiality image).
- <<t_1>>((<<t_2>>(conf_text P emphasis))∨(<<t_3>> (conf_image P table))) expression is true for those documents which contain valid confidentiality text or image.
- <<t_1>>((<<t_2>>(conf_text P emphasis)) ∧ (<<t_3>> (conf_image P table))) expression is true for those documents which contain both valid confidentiality text and image.

- <<t_1>> ((<<t_2>>(conf_text P ¬emphasis))∨(<<t_3>> (conf_image P ¬table))) expression is true for those documents which contain at least one non-valid confidentiality note.

3.3 Evaluation algorithm

Using only model theoretical approach can be sufficient for analyzing properties of multimedia Web documents because a property can be represented by a TSDL expression, Web documents and transformations are represented as the model of the logic and evaluating an expression over a model clearly identifies if the property is hold or not. Consequently, the major question is that what kind of algorithms can be provided to compute the truth of an expression over a model. Similar problems usually arise in model checking. The problem can be divided into two questions. Firstly, a document level approach will be presented to evaluate document level expressions over document level models. Secondly, a transformational level algorithm will be presented with the contribution of document level one.

Document level algorithm is based on a relational algebraic approach. Since expressions are true for some nodes of the document model, therefore a document level expression can be regarded as a subset of nodes of the original document model. First of all, partial maps are stored as binary relations: $V\_P \subseteq V \times V$ for *p* parent map, $V\_C \subseteq V \times V$ for *c* children map and $V\_AP \subseteq V \times AP$ for *ap* atomic predicate map where V is the set of nodes of the document model. Secondly, computational expressions must be associated with the logical operators. These computational expressions determine the set of nodes for which the expression is true. Let us consider that we have two expressions, e_1 and e_2, subsets of nodes of the model for which these expressions are true: V_{e1}, V_{e2}. We can compute the subset of nodes for which $e_1 \wedge e_2$ is true by making an intersection of the two sets ($V_{e1 \wedge e2} = V_{e1} \cap V_{e2}$). Similarly, disjunctions can be computed by union.

Table 1. Document level logical operators and the associated relational algebraic expressions.

Logical expression	Relational algebraic expression
‘a’ (atomic predicate)	$\sigma_{AP=a}(V)$
T	V
⊥	∅
¬	$V - V_e$
∧	$V_{e1} \cap V_{e2}$
∨	$V_{e1} \cup V_{e2}$
P	$\pi_V((V\_P) \cap (V_{e1} \times V_{e2}))$
$C_\exists$	$\pi_V((V\_C) \cap (V_{e1} \times V_{e2}))$
$C_\forall$	$\pi_V((V\_C) \cap (V_{e1} \times V)) - \pi_V(((V\_C) \cap (V_{e1} \times V)) - ((V\_C) \cap (V_{e1} \times V_{e2})))$

Table 1. lists all the relational algebraic expressions that are associated with document level logical expressions[2] (σ denotes selection, π_V is a projection to the first set of a binary relation). Further information about relational algebra can be found in [14].

Unfortunately, TSDL expressions cannot be computed similarly because transformational level models are usually infinite. Therefore, a tableau based method is applied. Complex expressions are decomposed to atomic expressions by applying syntactic rewriting rules. Atomic expressions are document level expressions so they can be evaluated by the previous relational algebraic approach. Using the syntactic rewriting rules on Figure 3 with a depth-first search strategy, a simple evaluation algorithm is presented.

Evaluating a document level expression requires polynomial time and space complexity, because relational algebraic operators can be evaluated in polynomial time, and space is only required for storing the relations (which is polynomial in the size of nodes). Unfortunately, the situation is not so good at transformational level. Syntactic rewriting has branching at disjunctions. Consequently, time complexity of evaluating a TSDL expression can be exponential in the size of the number of TSDL level disjunctions. At practical applications, the number of disjunctions are not extremely high, so time complexity remains acceptable. Applying depth-first search at syntactic rewriting, space complexity of the algorithm remains polynomial.

$$\frac{TM,\ d \models TSDL_1 \wedge TSDL_2}{TM,\ d \models TSDL_1\ ,\ TM,\ d \models TSDL_2}$$

$$\frac{TM,\ d \models TSDL_1 \vee TSDL_2}{TM,\ d \models TSDL_1} \qquad \frac{TM,\ d \models TSDL_1 \vee TSDL_2}{TM,\ d \models TSDL_2}$$

$$\frac{TM,\ d \models <<t_1;t_2;t_3;..;t_N>>\ TSDL}{TM,\ t_N(\ldots t_2(t_1(d))) \models TSDL}$$

TM, d |= TSDL, if TSDL is a document level expressions, and there is an ‘x’ node of ‘d’ document model for which d, x |= exp.

Fig. 3. Tableau based algorithm for evaluating a TSDL expression over a transformational model. The algorithm is a simple modification of the tableau based algorithm of Hennessy-Milner logic, which is primarily used in model checking [15].

[2] Proof of the correctness of relational algebraic expressions is being published in Periodica Polytechnica.

4 Conclusion and Further Work

This paper summarizes some theoretical foundations of Transformational Structured Document Logic, which is a logical methodology for analyzing properties of XML or HTML documents containing multimedia elements. Beside theory, a shell architecture has also been implemented in Java to test the concepts between real circumstances. The architecture realizes an HTML and an XML parser, the algorithms for evaluating document level and transformational level expressions, and several basic atomic transformations.

In a sense, TSDL does not represent a new logic, because it is a subset of several more general logical frameworks. Document level of TSDL can be considered as a special kind of Propositional Dynamic Logic. For example, DIFR logic (a kind of Propositional Dynamic Logic) contains more syntactic elements than document level of TSDL does [16]. Transformational level of the logic realizes a special Hennessy-Milner (HM) logic which nodes are document level models, and there is only one modal operator at this level called deterministic execution of transformations. Deterministic execution is the same as all executions and some executions in HM, because executions are not necessarily deterministic in HM but surely deterministic in TSDL. However, these general logical frameworks usually do not concentrate on Web documents, instead they deal with actions or time [13,15]. Therefore, TSDL can be considered as a domain specific logic which syntax and semantics also focus on Web documents. This specialization causes several benefits. Firstly, syntax of the logic directly expresses the necessary formulas for the most common applications. Secondly, the limited approach entails several computational benefits which are primarily manifested in easy and relatively fast evaluation algorithms. Although this article covers mainly the model theory of the logic, the limited approach could result in significant simplification in proof theory.

TSDL can be extended and further studied from both practical and theoretical points of views:

- The efficiency of the evaluation algorithm can be further increased by using sophisticated data structures, like Hash tables or Hash trees [17].
- The applications of soft evaluation techniques would lead not only to the identification of the truth or falsity of a property of Web documents, but to the recognition of a more descriptive value as well.
- Pre-transformations, filters and multimedia transformations should be further investigated. At this point they are quite ad-hoc and implemented by Java objects. However, an ontology of multimedia transformations would also be useful.
- Proof theory of the logic should be developed. With proof theory, consequences or common properties of the truly evaluated expressions could be identified.
- Beside theoretical issues and experimental implementation, developing industrial application using TSDL remains an open question needing further investigation and research.

References

1. XQuery 1.0: An XML Query Language, W3C Working Draft, http://www.w3.org/TR/xquery/#nt-bnf, 2002.
2. Raymond Lau, Arthur H.M, A Logic-Based Approach for Adaptive Information Filtering Agents, Lecture Notes in Computer Science, Vol. 2112, pp.269-280. 2001.
3. R. Cooley and P. Tan and J. Srivastava, Websift: The Web site information filter system, In Proceedings of the 1999 KDD Workshop on Web Mining, San Diego, 1999.
4. M. Ackerman and B. Starr and M. Pazzani, The Do-I-Care Agent: Effective Social Discovery and Filtering on the Web, In RIAO '97: Computer-Assisted Information Searching on the Internet, pages 17--31, Montreal, CA, June 1997.
5. XML Base, W3C Recommendation, http://www.w3.org/TR/xmlbase/, 2001.
6. William I. Grosky and Yi Tao, Multimedia Data Mining and Its Implications for Query Processing, "{DEXA} Workshop 1998.
7. XML Path Language (XPath), Version 1.0, W3C Recommendation, http://www.w3.org/TR/xpath, 1999.
8. XML Path Language (XPath) 2.0, W3C Working Draft, http://www.w3.org/TR/xpath20/ 2002.
9. G. Conforti and G. Ghelli and A. Albano and D. Colazzo and P. Manghi and C. Sartiani, The Query Language TQL, Proc. of 5th International Workshop on Web and Databases (WebDB 2002), 2002.
10. K. Lodaya, R. Ramanujam, An Automation Model of User-Controlled Navigation on the Web, Implementation and Application of Automata, 5th International Conference, CIAA 2000.
11. Luca de Alfaro, Model Checking the World Wide Web, Lecture Notes in Computer Science, Volume 2102, pp. 337-350, 2001.
12. HTML 4.01 Specification, W3C Recommendation, http://www.w3.org/TR/html401/, 1999.
13. Imre Ruzsa, Introduction to Modern Logic, in Hungarian, Osiris Press 2000.
14. Jeffrey D.Ullman, Jennifer Widom, A First Course in Database Systems, Prentice Hall, Inc, 1997.
15. M. Müller-Olm, D. Schmidt, B. Steffen, Model-Checking: A Tutorial Introduction. Lecture Notes in Computer Science (LNCS), Vol. 1694, pp. 330--354. Springer-Verlag, 1999.
16. Giuseppe De Giacomo, Maurizio Lenzerini, PDL-based framework for reasoning about actions, In Lecture Notes in Artificial Intelligence n. 992, pages 103--114. Springer-Verlag, 1995.
17. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Introduction to Algorithms, MIT Press, 1990.

A Constraint Satisfaction framework in Document Recognition

Alexandre S. Saidi

Ecole Centrale de Lyon
Mathematics and Computer science Department
B.P. 163. 69134 Ecully - France
http://www.ec-lyon.fr - Alexandre.Saidi@ec-lyon.fr

Abstract. In this paper, the Grammatical inference (GI) is considered as an instance of the Constraint Satisfaction Problem (CSP), and a Constraint Logic Program (CLP) is proposed which instantiates automata in a lattice. In this task, the inference engine takes as input a set of individual positive and negative examples of documents and outputs a set of rules that recognises similar documents.
In the theoretical outline, the Pattern Recognition and the GI problems are considered in an algebraic framework in which a set of constraints will define the process of *generalisation*. The CLP implementation discussed here is used in a document handling project in which (paper) documents are typographically tagged and then recognised. The main aim is to extract the physical and the logical structures of a given set of (paper) documents in order to produce a machine readable form like XML, HTML or TEX format.

1 Introduction

This paper describes the related work in a Pattern Recognition project applied to documents like summaries, dictionaries, scientific reports, bibliographic basis, encyclopedia and so on from examples. The main purpose is to extract the hierarchical structure and the content of these classes of documents. As such, we apply the process of Grammatical Inference to a set of regular production rules. These rules represent the logical structure of these documents (i.e. text and images organised in titles, chapters, sub-chapters and so on in the case of summaries) and are defined for each element of a sample set. Negative examples can be provided in order to denote structures that should be rejected. The inference engine then produces a representative grammar that will reorganise texts and images in their respective contexts.

As suggested in [5], the problem of grammatical inference can be considered in a CSP framework (see e.g. [3]). Although some work ([10]) tackled this problem as an instance of graph colouring, the proposed approach gave only the specification of a (language inclusion) lattice giving a quite general idea with no CLP (see e.g. [2]) framework.

In ([7]), Gold showed that any recursively enumerable class of language is identifiable using a complete representation with positive (I_+) and negative (I_-) data. . Hence, the class of regular languages cannot be correctly identified from only positive examples. Although the usual case in document handling is to learn from only positive examples (I_+), the result (a set of production rules) can be drastically refined by negative examples (I_-) and avoid *over generalisation*.

It is known that any algorithm that would construct a DFA (deterministic finite automaton) with a minimum number of states compatible with all the data already processed can identify any regular language in the limit ([6]). The search space being a (language inclusion) lattice ([8]), we develop here an original and complete algebraic framework for the Grammatical Inference and present a relation on this search space that characterises the construction of partitions over the lattice of automata represented by $I=(I_+ \cup I_-)$.

To realize that, an initial algebra A_{G_I} is assigned to the regular grammar G_I of the sample I. Then the main result is the definition of a quotient algebra $A_{G_I/R}$ that leads to an uniquely defined isomorphism from $A_{G_I/R}$ to the language of the (to be generated) automaton A.

Within the algebraic framework, we discuss a general Constraint Satisfaction specification that characterises the search space of the GI problem and then define a set of constraints that will construct the final DFA. The results are used, among others, in a project on paper document processing whose one application is the translation of the documents into HTML/XML/TEX format.

2 Regular Inference

The Inductive Inference paradigm is the basis of the automatic learning problem. In the Syntactical Pattern Recognition (see e.g. [17]), many grammatical inference algorithms are proposed that are used in the learning step of the pattern recognition tasks ([14], [15], [16]). In order to correctly identify regular languages, positive and negative examples are to be provided to represent the language to be learned.

Example 1. From scientific reports, we may have $I_+=\{r_1, r_2, r_3\}$, $I_-=\{r_4, r_5\}$:
r1 : report $\leftarrow abstract, ackn glmnt, outline, chapter, chapter, references.$
$r2 : report \leftarrow abstract, outline, chapter, subchapter, chapter, references, index.$
$r3 : report \leftarrow abstract, acknglmnt, outline, fig\_table, chapter, chapter, index.$
$r4 : report \leftarrow abstract, outline, references.$
$r5 : report \leftarrow abstract, acknglmnt, chapter, index.$

Here, r_4 denotes that there is no well formed *report* without any *chapter* while r_5 rejects a *report* without *outline*. Hence, in this case, saying that an *acknglmnt* (acknowledgement) will always follow an *abstract* is not a usable fact.

In this paper, first we give some basic definitions for reference. Then, in the section 4, an algebraic specification of the GI problem is developed. In the sections 5, 6 and 7, some practical issues, the implementation of the proposed CSP framework together with some examples are reported. Then, some relationships with other works in the field is recalled in the section 8.

3 Basic notations and definitions

We assume that the reader is familiar with the basic notions in the grammars (see e.g. [13]).

A *finite automaton* (FA) A is a quintuplet (Q, Σ, δ, q_0, F) where Q is the set of states, Σ is the set of input symbols, δ : Q x $\Sigma^* \rightarrow 2^Q$ is the transition function, $q_0 \in$Q is the start state and F $\subseteq$ Q is the set of final states.

For an automaton A, the language accepted by A is denoted by $L(A)$. A language is *regular* iff it is accepted by a FA. An automaton A is *deterministic* (DFA) if for all q$\in$Q and for all a$\in$ Σ, δ(q,a) has at most one element.

If A=(Q, Σ, δ, q_0, F) is a FA and π a *partition* of Q, B(q,π) is the only *block* (result of the fusion of states) that contains q and we denote the quotient set by Q/π as the set of all partitions {B(q,π) | q$\in$Q}. If A is a FA and a partition π over Q, the *quotient* (or derived) automaton A/π = {Q/π, Σ, δ', B(q_0,π), {$B_i \in$ Q/π | $\exists$ $q \in$ B_i, q$\in$F} where δ' is defined by $\forall$B,B'$\in$Q/π , $\forall$a$\in$ Σ, B'$\in$ δ'(B,a) iff $\exists$ q,q'$\in$Q, q$\in$B, q'$\in$B' : q'$\in$ δ(q,a).

It is easy to see that for a partition π over Q, L(A) $\subseteq$ L(A/π). The set of all automata derived from A is a (language inclusion) lattice $Lat(A)$.

A *context-free grammar* (CFG) is denoted by G=(N, T, P, S) where N and T are finite sets of *non terminals* and *terminals* and P is a finite set of productions. The special non terminal S is called the *start* symbol.

A (right) *regular grammar* is a context-free grammar whose the productions rules are of the form A $\rightarrow$ α. or A $\rightarrow$ α B. where α $\in$T, A,B$\in$N.

The *language* L(G) is any string ω $\in$T* such that there is a derivation from S to ω (denoted by S $\Rightarrow^*$ ω). By extension, the language of any non terminal A$\in$N is any string μ such that for τ,σ,μ $\in$T*, S $\Rightarrow^*$ τA$\sigma$$\Rightarrow^*$ $\tau\mu\sigma$.

Given I_+ the positive representation from a regular language L, I_+ is said to be *structurally complete* if all transitions of (the unknown) automaton $A(L)$ are used in the acceptance of strings in I_+ terminating in a final state in F_+ (the set of the final states from I_+). This notion is extended to I_- given that $I_+ \cap I_- = \emptyset$, $F_+ \cap F_- = \emptyset$, F_- is the set of final states of I_-, I_- is structurally complete with respect to its own language. The language L(A) of I denotes only those accepted strings using a final state in F_+ and rejecting those of I_-.

Let us recall some results in the field of regular grammatical inference. Even though we do not use these results in this paper, it is usefull to have them in mind and the sections that follow will outline some results similar to them.

If $A(L)$ is the *minimal (resp. maximal) canonical automaton* of a language L, then A is a DFA with the minimum (resp. maximum) number of states accepting L. The *maximal canonical automaton* is denoted MCA. One can define the *prefix tree* acceptor of I denoted by *PT(I)* from the MCA by merging states sharing the same prefix. If I is a structurally complete sample of a regular language L, then there exists a partition π over the states of PT(I) such that PT(I)/π is isomorphic to A(L). If I is a positive sample, the set Γ of automata such that I is structurally complete with any automaton in Γ is *Lat(MCA(I))*. If L is the target language and I is structurally complete with respect to A(L), then A(L) is an element of *Lat(PT(I))*. The automaton A(L) is learnable as a partition PT(I)/π isomorphic to A(L) in the lattice of all possible partitions PT(I)/π. Note that this search space grows exponentially with the size of the state set in $PT(I)$ and therefore with the size of I.

In the algebraic specification below, the properties of partitions over the algebra A_{G_I} associated to the grammar G_I of I are depicted and we formally characterise a relation from these partitions to L(A). This is done by the definition of a set of constraints defining relations over the terms of A_{G_I} that produces the quotient-algebra whose terms are isomorphic to those of L(A).

Quotients of the A_{G_I}-algebra giving a (language inclusion) lattice, our *main aim* in the Grammatical (regular) Inference described here is to characterise this lattice and to guide the search in it.

4 An Algebraic specification of the grammatical inference

The Grammatical Inference problem can be specified by using the relation between an initial many sorted algebra and grammars ([11]). This relation over context-free grammars is easily extended to the regular grammars. Such a grammar is used as a (rewrite) system to define an algebra. The major property is that the defined algebra is *initial* in the category of the same signature.

To construct the algebra associated to a context-free grammar G, each non terminal of G is assigned to a class of derivation tree. Consequently, the non terminals of G are sorts of a many sorted algebra whose operations are defined by the production of G. The derivation tree (and hence the language) of any non terminal X denotes the carriers of the sort X of the algebra.

Let G=(N,T,P,S) be a context-free grammar and L_G be its language. Let be associated to G the A_G-algebra whose signature is ((N$\cup$T), Op) where Op is the set of names given to the productions in P. The terms of this algebra are derivation trees starting from any non terminal of G. A_G can be extended if we consider the elements of N as the ranked variable radical (χ=\{ν_i $|\nu \in$ N, i is a natural number\}) and define an equational theory together with a well-formed substitution and a consistent replacement function over χ in A_G.

The terms of the A_G-algebra are derivation trees constructed by using the names of the productions of G. Constants (0-air operators of the A_G-algebra) are terminals of G. By *equivalence*, we mean an equivalence relation R over the sorts of A_G. Since the elements of N are sorts of A_G-algebra, any string ω such that X$\in$N, X$\Rightarrow^*$ ω is typed by X, particularly when ω $\in$L(G) where for S the start non terminal, S$\Rightarrow^*\omega$.

An A_G-algebra is *initial* in a category C based on the same signature if for all algebra B of C, there exist a unique homomorphism f: $A_G \rightarrow$ B ([12], [11]).

We may define a homomorphism f : $A_G \rightarrow$ *sem* where *sem* can denote the language L(G) and hence to associate a *string semantics* to the terms of A_G. This function can be a syntax-directed translation (or compositional semantics) one. Here, considering the sample set I (and its automata) and L(A) the language of A (the final DFA to be constructed), we are interested in f : $A_{G_I} \rightarrow L(A)$.

Let the A_{G_I}-algebra be associated to the grammar G_I of the set of samples I=$I_+ \cup I_-$ where $I_+ \cap I_- = \emptyset$. The terms of this algebra are derivation trees for elements of I. Consider the set of finite automata associated to elements of I and let ***Tree(I)*** denote the tree of all theses automata. Assume that the (no deterministic but not ambiguous ϵ-free) grammar G_I=(Q∪{S}, Σ, P_δ, S) where P_δ is the set of the names given to the transitions in *Tree(I)*, is associated to *Tree(I)* with the start symbole : S → p_{01} | p_{02} | ...

Let L_{I+} be the language of the positive examples (resp. L_{I-} for the negative ones) generated by the final automaton A. Then, for any partition of Tree(I) where some automaton are considered *equivalent*, $I_+ \subseteq L_{I+}$ and $I_- \subseteq L_{I-}$. We have $L_{I-} \subseteq \Sigma^*$ - L_{I+} and $L_{I_-} \cap L_{I_+} = \emptyset$. The section 8 discusses how and why we may also set $I_- \subseteq \Sigma^*$ - L(A) or $L_{I+} \subseteq L_I$.

Let G_I be the (regular) grammar associated to Tree(I), R a congruence relation, a partition Tree(I)$_{/R}$ from Tree(I) and its regular grammar G_R. Let A_{G_I} and $A_{G_I/R}$ be the algebra assigned to G_I and G_R. In the following section, we will define a homomorphism $homo_R$ from A_{G_I} to $A_{G_I/R}$. Then, we will state a constraint satisfaction specification of the (language inclusion) lattice induced by $homo_R$ and propose a CLP program that will search, under some constraints, for a (not necessairly minimal canonic) DFA in that lattice.

4.1 The quotient algebra

Let A_{G_I}=((Q ∪ Σ), Op) be an algebra associated to the (regular) grammar of the sample set I. Terms of A_{G_I} are derivation trees (let note them by $\hat{a}$ or $\hat{b}$) of the form *ri(rj, rm(..., rk(rn)...)* and of some sort q ∈ Q. Let R a congruence relation on A_{G_I}. Op is the set of names (like *ri*) of rules of G_I of the form *(q', α → q)* or *(α → q)*, $\alpha \in \Sigma$, q,q' ∈ Q. The quotient algebra induced by R is defined by $A_{G_I/R}$ = ($\overline{(Q \cup \Sigma)}$, $\overline{Op}$) with :

1- $\overline{(Q \cup \Sigma)}$ = {[$\hat{a}$] | $\hat{a}$ a derivation tree whose type is q ∈ Q} where the congruence class [$\hat{a}$] is defined by [$\hat{a}$]={$\hat{b}$ a derivation tree of type q ∈ Q | ($\hat{a}$,$\hat{b}$) ∈ R};
2- $\overline{Op}$=set of $\overline{ri}$ for each element of Σ, if *ri* is the name of a production rule of the form $\alpha \rightarrow q$ with q ∈ Q and $\overline{ri}$: [α] → [q];
3- $\overline{Op}$= set of $\overline{ri}$: ([q'], [q"]) → [q] if *ri* is the name of a production rule *(q', q") → q* with q, q', q" ∈ Q is defined by
$\overline{ri}$($\overline{rj}$,$\overline{rm}$(..., $\overline{rk}$($\overline{rn}$)...) = [ri(rj, rm(..., rk(rn)...)]. A derivation tree [$\hat{a}$] in $A_{G_I/R}$ is constructed using elements congruent to $\hat{a}$∈A_{G_I}. □

Although a term $\hat{a}$ of A_{G_I} is like *ri(rj, rm(rl, ..., rk(rn)...)* with r_i, r_j, ... ∈ Op, for the sake of clarity, we will rewrite $\hat{a}$ by *ri(α, rm(β,..., rk(γ)...)* when *rj* (resp. *rl*, *rn*, etc.) is the name of a production rule like $\alpha \rightarrow q$. This is also motivated by the fact that [α] denotes the equivalence class of the constant α whenever [$\hat{a}$]=[α]. We set [α]=α for each $\alpha \in \Sigma$.

Operations of $\mathrm{A}_{G_I/R}$ are well defined since R is reflexive, symmetric, transitive and compatible such that from $[\hat{a}]=[\hat{b}]$ we conclude that $(\hat{a},\hat{b}) \in$ R. Thus, $((A_{G_I})\ \mathrm{r}_{i1}(\alpha 1,\ \mathrm{r}_{j1}(\alpha 2,\ ...,\ \mathrm{r}_{k1}(\alpha \mathrm{n})...)\ ,\ (A_{G_I})\ \mathrm{r}_{i2}(\beta 1,\ \mathrm{r}_{j2}(\beta 2,\ ...,\ \mathrm{r}_{k2}(\beta \mathrm{n})...)) \in \mathrm{R}$ implies $[(A_{G_I})\ \mathrm{r}_{i1}(\alpha 1,\ \mathrm{r}_{j1}(\ ...,\ \mathrm{r}_{k1}(\alpha \mathrm{n})...)] \equiv [(A_{G_I})\ \mathrm{r}_{i2}(\beta 1,\ \mathrm{r}_{j2}(...,\ \mathrm{r}_{k2}(\beta \mathrm{n})...)]$. Equivalently, $(\mathrm{r}_i,\ \mathrm{r}_j) \in$R implies $\overline{r_i} \equiv \overline{r_j}$ (same as $[\mathrm{r}_i] \equiv [\mathrm{r}_j]$).

Quotient algebra are characterised by the universal property (up to an isomorphism [12]). This property is stated by the following (homomorphism) theorem applied to A_{G_I} :

Theorem 1. *Let A_{G_I} associated to Tree(I) be the algebra and R a congruence relation on A_{G_I}. Then $homo_R : A_{G_I} \to A_{G_I/R}$ defined by $homo_R(\hat{a}) = [\hat{a}]$ for $\hat{a} \in A_{G_I}$ is a homomorphism that has the following property.*

Let f : $A_{G_I} \to$ L(A) a homomorphism with the (former) congruence relation R [1] *, then there exists a unique homomorphism $\bar{f}$ such that the following diagram of mapping is commutative, i.e., $f = \bar{f} \circ homo_R$.*

$$\begin{tikzcd}
\mathrm{A}_{G_I} \arrow[rr, "f"] \arrow[dr, "homo_R"'] \arrow[rr, phantom, shift right=4ex, "="] & & L(A) \\
& \mathrm{A}_{G_I}/\mathrm{R} \arrow[ur, "\bar{f}"'] &
\end{tikzcd}$$

Let us show first that $homo_R$ is a homomorphism before proving the theorem 1.

Theorem 2. *$homo_R$ is a (quotient-)homomorphism.*

Proof. By the definition of the quotient algebra $\mathrm{A}_{G_I/R}$, for the constant symbols where r_i in Op is the name of the rule $\alpha \to q$ we have $homo_R(r_i) = \overline{ri} = [ri]$ and, for operations r_i in Op where r_i is the name of a production rule *(q', α)* $\to$ q, we have :

$$\begin{aligned} homo_R(\mathrm{ri}(\mathrm{rj},\ \mathrm{rm}(\mathrm{rl},...,\ \mathrm{rk}(\mathrm{rn})...)) &= [\ \mathrm{ri}(\mathrm{rj},\ \mathrm{rm}(\mathrm{rl},...,\ \mathrm{rk}(\mathrm{rn})...)\ ] \\ &= \overline{ri}(\overline{rj},\overline{rm}(\overline{rl}\ ...,\ \overline{rk}(\overline{rn})...). \quad \square \end{aligned}$$

Thus $homo_R$ is a homomorphism and we have (trivially) :

Lemma 1. *the value of a derivation tree â in $A_{G_I/R}$ is the equivalence of â in A_{G_I}. That is : $((A_{G_I/R})\ (\hat{a})) = [(A_{G_I})\ (\hat{a})]$.*

Proof. this is an immediate consequence of the quotient algebra $\mathrm{A}_{G_I/R}$. Given The above properties of the relation R, we naturally have $homo_R(\hat{a}) = [\hat{a}]$. □

Now, we can give the proof of the above **theorem-1**

[1] with R $\subseteq$ Eq(f) which is the congruence induced by f with Eq(f)= $\{(\hat{a},\hat{b})\ |\ \hat{a},\hat{b} \in \mathrm{A}_{G_I}$ and $\mathrm{f}(\hat{a}) \equiv \mathrm{f}(\hat{b})\}$ giving congruent words in L(A), e.g. possibly f(”$\widehat{abbbc}$”) $\equiv$ f(”$\widehat{abbbbbc}$”) where $\widehat{\alpha}$ is the derivation tree of the word $\alpha \in \Sigma$*.

Proof (of the theorem-1). Let $\bar{f}$: A$_{G_I/R}$ → L(A) be defined by : $\bar{f}$([$\hat{a}$]) = f($\hat{a}$), $\hat{a}$ is a term of A$_{G_I}$. We have :

⋆ $\bar{f}$ is uniquely defined : if $\bar{f}$ defines a homomorphism at all, then $\bar{f}$ is unique since every other homomorphism g : A$_{G_I/R}$ → L(A) with *f= g ∘ homo$_R$* must satisfy g([$\hat{a}$]) = f($\hat{a}$).

⋆ $\bar{f}$ is well-defined : for $\hat{a}$,$\hat{b}$ ∈ A$_{G_I}$, ($\hat{a}$,$\hat{b}$) ∈ R (and, by the definition of f where ($\hat{a}$,$\hat{b}$) ∈ Eq(f)), we have f($\hat{a}$) ≡ f($\hat{b}$) which implies that $\bar{f}$([$\hat{a}$]) ≡ $\bar{f}$([$\hat{b}$]). Hence $\bar{f}$ is well defined.

⋆ $\bar{f}$ is a homomorphism : Let *ri* ∈Op the name of a the operation $\alpha \rightarrow q$ denote a symbole $\alpha \in \Sigma$. Then we have $\bar{f}(\overline{ri})$=$\bar{f}$([ri])=f(ri)=α ∈L(A) since f is a homomorphism.
Otherwise, for *ri* in Op be the name of the operation *(q', q") → q* with q, q',q" ∈ Q, we have for [$\hat{a}$] in A$_{G_I/R}$ (the dot symbol is the monoid concatenation):

$$\begin{aligned}\bar{f}(\overline{ri}(\overline{rj}, \overline{rk}(...(\overline{rn}))) &= \bar{f}(\text{[ri(rj, rk(rl,...,rv(rn)))]}) \\ &= \text{f(ri(rj, rk(rl,...,rv(rn)...))} \\ &= \text{(L(A)) f(rj)} \cdot \text{f(rk(rl,...,rv(rn))...) and by the definition of } \bar{f} \\ &= \text{(L(A))}\ \bar{f}(\overline{rj}) \cdot \bar{f}(\overline{rk}(\overline{rl}),..., \overline{rv}(\overline{rn})))...) \quad \square\end{aligned}$$

Thus, $\bar{f}$ is a homomorphism.

Note that if we make $R = Eq(f)$ for each sort q∈Q, then $\bar{f}$ is injective and two equivalent terms of A$_{G_I}$ using the equivalence class [q] will generate equivalent (sub) words of the type q. Furthermore, if I is structurally complete and represents L(A) and if G$_I$ (and hence G$_R$) are not ambiguous, then f is surjective. Hence, $\bar{f}$ is an isomorphism. Note that the problem of the ambiguity and the question of emptyness of the intersection of regular languages are decidable [13]. This result is similar to the existence of a partion PT(I)/π isomorphic to L(A) used in the literature. Furthermore, the family of the relation R defines the quotient algebra which characterises the (language inclusion) lattice **Lat**$_R$ (similar to Lat(PT(I)) in the literature).

We are looking for a method that, under some hypotheses, gives L(A) et hence, will complete *a posteriori* the set I of samples. The set I must be structurally complete and be representative of L(A) since we want that all equivalence classes be calculated giving A$_{G_I/R}$. Hence, if we want to have the above properties for R, *homo$_R$*, $\bar{f}$ and f, then there must be no [$\hat{a}$] in A$_{G_I/R}$ other than those given by R over A$_{G_I}$. Consequently, the following propositions hold.

Proposition 1. *I is structurally complete and representative of L(A) if by the homomorphism homo$_R$, all equivalence classes of $\hat{a}$∈A$_{G_I}$ are calculated in A$_{G_I/R}$ (and there exists no other).*

Proposition 2. *Given I structurally complete and representative of L(A), a family of congruence relation R constructs a class (i.e. the category of sig-A$_{G_I}$-algebra) of the initial algebra A$_{G_I/R}$ giving the (language inclusion) lattice Lat$_R$ where for the final automaton A of L(A), A is an element of Lat$_R$.*

In the next section, we will define a CSP specification by a CLP predicate that defines the congruence relation R and hence charecterises the search space Lat$_R$ and the instanciations in it. Then we will discuss the properties of I$_+$ and I$_-$ with respect to L$_I$, L$_{I_+}$, L$_{I_-}$ and L(A).

5 The Definition of R : the Congruence predicate

Recall that R is a congruence relation over (the sorts of) A$_{G_I}$. Let $\hat{a}_1$, $\hat{a}_2 \in$ A$_{G_I}$
where $\hat{a}_1$ = r$_{i_1}$(α_1, r$_{i_2}$(α_2, r$_{i_3}$(α_3, r$_{i_4}$(α_4,..., r$_{i_{n-1}}$(α_{n-1}, r$_{i_n}$(α_n)...)
and $\hat{a}_2$ = r$_{j_1}$(β_1, r$_{j_2}$(β_2, r$j3$(β_3, r$j4$(β_4,..., r$_{jn-1}$(βn-1, r$_{jn}$(β_n)...).

The Congruence predicate constructs the store θ (a set of constraints) and assigns an equivalence class to each q$_i$∈Q. The set θ may contain constraints like x *in* r, = and ≠. Whenever the set of final constraints is satisfiable, if there is more than one solution, then we will choose the one which minimises the number of equivalence calsses. Initialy, $\theta = \emptyset$.

In order to extract equivalence classes, this predicate is applied to every pair of (compound) terms of A$_{G_I}$. Within each couple of terms, the predicat is applied to every couple of sub-term of $\hat{a}_1$ and $\hat{a}_2$. Backtraking is used to compute a consistent θ (which denotes quotient algebra and Lat$_R$). Initialy, [q$_i$] is the equivalence class of each q$_i$∈Q. Elements of I$_+$ and I$_-$ are distinguished, hence we recognise final states (F$_+$ and F$_-$ with F=F$_+$ ∪ F$_-$ and F$_+$ ∩ F$_-$=∅) of these two sets from each other and from any other equivalence class.

Predicate congruence(t$_1$, t$_2$) adds constraints to θ
Let t$_1$ and t$_2$ be partial trees (partial composed terms) of $\hat{a}_1$,$\hat{a}_2$
t$_1$ = r$_{i_1}$(α_1, r$_{i_2}$(...)...) and t$_2$ = r$_{j_1}$(β_1, r$_{j_2}$(...)...) which denote
r$_i$: α × p'$_1$ → p$_1$ r$_j$: β × p'$_2$ → p$_2$

(1) if $p1$ and $p2$ are different final states in ($F+ \times F-$) then set [p1]≠[p2].
(2) if $\alpha = \beta$ then set ([p'1]= [p'2] ⇒ [p1] = [p2]) *(preserves the DFA condition)*
(3) if $\alpha \neq \beta$ then set [p1] ≠ [p2]

Remarque : in (2), we set the determinism constraint for all automata in the lattice. in (3), we may equivalently have {*set* $\alpha \neq \beta \Rightarrow$ *[p1]* ≠ *[p2]*} or {*set [p1] = [p2]* ⇒ $\alpha = \beta$*)*} but this formulation makes a strong dependence between cases (2) and (3). Nevertheless, this last formulation shows why we do not need to start with $PT(I)$ since it sets equivalence classes by *([p1] = [p2]* ⇒ $\alpha = \beta$*)* and we choose the final automaton with the fewest states. Also, the case (3) sets constraints such that for an equivalence class [q_i], all prefix will be in the same equivalence class [α]. This is an application of the well known *pumping lemma* (see e.g. [13]) and makes the final automaton ϵ-free (we currently study the properties of equivalence class [α] in the case of the grammatical inference of context-free grammars). Furthermore, the case (3) sets two different equivalence classes if they are both states from F$_+$ × F$_-$. Recall that A(L) only accepts elements of L(A) whose derivations terminate with a final state in F$_+$.

For a given instantiation of the equivalence classes, the set of constraints θ defines a set of equivalence classes for $q_i \in Q$ giving a partition and hence a quotient algebra. The set of all these partitions is Lat_R.

The demonstration of the soundness, the completeness and the termination (also the existence and the uniqueness) properties of the Congruence predicate are out of the scope of this paper even though they are quite straightforward.

Given the case (3), we exclude equivalence classes where the prefix symbol $[\alpha]$ are from different classes. Hence, we will not have unsound final states with the possibility to continue a derivation from these states. The final sound DFA is minimal given the set of constraints on it. It is not necessarily canonic since this property does not bring any advantage here and *a contrario*, in some cases, will generate an unusable language (see also the use of tables in section 8).

6 Realisation in GNU-Prolog

We used Gnu-Prolog ([4]) to encode the Congruence predicate. Gnu-Prolog is CLP environment with finite domain solver. We can set integer, boolean and reified constraints together with some others such as symbolic constraints. The enumeration of the values in a domain is possible with minimisation of some function. Gnu-Prolog implements partial and full AC constraint propagation algorithms (see e.g. [2], [4]). We give here some details of the CLP realisation.

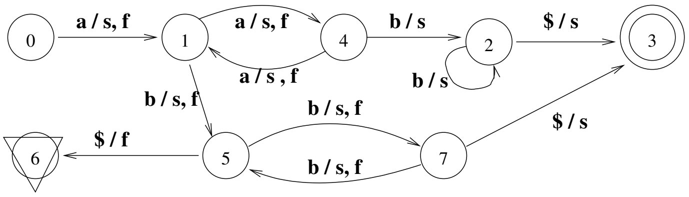
Every word of the sets I is extended with a special symbol $\$ \notin \Sigma$. We then associate one automaton A_i to each element of I. Each automaton A_i begins with its q_{i0} state (from the rule r_{i0}). Given this status of q_{i0} and the use of the \$ symbol, the output state of each transition is an input state of another one (except for the final states). Each transition is of the form *q=δ(α, q', code)* where *code* denotes the set (I_+ or I_-) and F_+ or F_- states when α=\$. The use of PT(I) is not necessary even though it is handled. Given the constraint (3) in the Congruence predicate, it is trivial to show that we will have the same final DFA no matter if we start from PT(I) or from Tree(I).

Each automaton has contiguous states numbers. If there is no I_- available, the final automaton generates only L_+ but may suffer from the *over generalisation* (see e.g; [8]). The final automaton A is then translated to a DCG ([1]) giving directly an operational grammar which begins to verify that elements of I_+ (resp. I_-) are accepted (resp. rejected).

7 Experimental Results

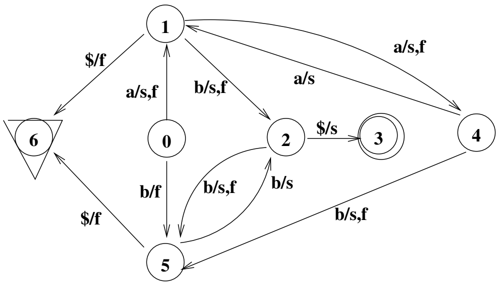
To validate the theoretical aspect of the related work, the implementation was tested using the experimental protocol cited in [9]. Within some fifteen regular languages (L1 to L15) proposed to evaluate a GI method, some are trivial like L_1=*a** or L_2=*(ab)**. Below, we present three of them completely. Note that the Example-3 is an original one. According to [6], sample sets (specially I_+) are (experimental) fixed point sets. We tested the other languages of the protocol with success and for each one, we generated an automaton with the fewest number of states given the constraints expressed in the Congruence predicates in the section 5. Results of these tests are available from the author. We also plan to implement this CSP framework more efficiently in an imperative language (like C++) with the help of the CSP libraries (see e.g. http://www.hulubei.net/tudor/csp/). Then, randomly generated large sample sets must be considered before we claim the complet practical usability of the implementation for any regular language.

Example 2. Consider the case of Σ={a,b}, L(A)=a$^+$b$^+$ where an odd number of *b* can not follow an odd number of *a*. We consider the following sample set for I$_+$ = {aab, aabb , abbbb , abb, aaabb, ...}, I$_-$ = {ab, abbb ,aaab , aaabbb, ...}. The resulting automaton is :



In this automaton, for a transition $\alpha/X, Y$, we have X=s (success) if X is present and Y=f (fail) if present. That is, the transition is a part of a possible derivation in L$_+$ (if success) and in L$_-$ (if fail). However, the user can constrain the system to generate the language of L$_+$ or of L$_-$ with or without the possibility that I$_-$ *enriches* L$_+$ and vice versa. The *enrichment* notion may be explained as follows. Suppose that [q] is originated from I$_-$. If there is any successful derivation of ω in L(A) using L([q]), then we say that I$_-$ *enriches* L$_+$. This situation is the case in the following example.

Example 3. Here, we consider I to denote the regular language $\mathbf{a}^n\mathbf{b}^m$ where n, m are both even or both odd. As one may remarque from the set I, the idea came from the analysis of $\mathbf{a}^n\mathbf{b}^n$ which is a CF language. We have, Σ={a, b} with I$_+$ = {ab, aabb, aaabbb, aaaabbbb...} and I$_-$ = {a,b, aab, abb, baa, bab, ...}. The generated DFA is :



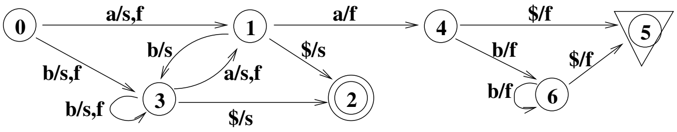
The language generated by this automaton is

L$_+$ = {a (aa)* b(bb)*} ∪ {aa (aa)* bb(bb)*}
L$_-$ = {aa (aa)* b(bb)*} ∪ {a(bb)$^+$} ∪ {b(bb)$^+$}

The enriched language is evident from the automaton. For example, "bb" derived from $q_0q_5q_2q_3$ is in enriched L$_+$ if we do not constrain all derivations to use only success ($< s >$ or $< s, f >$ tag) edges.

Example 4. In this example, we consider the regular language over the alphabet {a,b} without more than two consecutive a's. The sample set considered is :

I$_+$ = {a, ab, abbb, b, bb, ba, aba, abab, ...}, I$_-$ = {aa,aabb,bbaa,baab, ...}. The resulting DFA is :



The language generated by this automaton is :
L$_+$ = L$_{0+}$ = (a L$_{1+}$ | b L$_{3+}$) where sentences terminate with q$_2$ and
L$_{1+}$ = (L$_{2+}$ | b L$_{3+}$) , L$_{3+}$ = (b* L$_{2+}$ | b*a L$_{1+}$) , L$_{2+}$ = $\epsilon_{success}$
L$_-$ = L$_{0-}$ = (a L$_{1-}$ | b L$_{3-}$) where sentences terminate with q$_5$ and
L$_{3-}$ = b*a L$_{1-}$, L$_{1-}$ = (b L$_{3-}$ | a L$_{4-}$) , L$_{4-}$ = (L$_{5-}$ | b L$_{6-}$),
L$_{6-}$ = (L$_{5-}$ | b* L$_{5-}$), L$_{5-}$=ϵ_{fail}.
The enriched L$_+$ language is the same as L$_+$ since there is no fail edge (f tag) that can terminate with a final state in F+.

8 Discussion

Given the Congruence predicate, we can show that L$_{I-}$ ∩ L$_{I+}$ = ∅ and L$_{I-}$ ⊆ Σ* - L$_{I+}$. Meanwhile, the following is an open question : if an equivalence class [q] with its language L([q]) is generated by I$_-$, does any successful derivation of ω using L([q]) is in L(A).

One can use the resulting DFA in one or both of the following cases :
1- let I$_-$ enrich I$_+$. This means that (as in some learning process like the linguistic learning one), fails are used to learn successful experiences. Hence, L$_+$ is *wider* than if we would have used only I$_+$;
2- let I$_+$ enrich I$_-$. Hence, L$_{I+}$⊆L$_I$, that is, L(A) is somewhat reduced since L$_-$ is extended and L(A) ⊆ Σ* - L$_-$.

Recall that the final autumaton is an element of the lattice Lat$_R$. The bottom element of Lat$_R$ is the empty automaton and the top element is Σ*. We may restrict Lat$_R$ and consider Σ* as its top element but keeping $Tree(I)$ (or $PT(I)$) as its bottom element. The aim of the GI is to find a class of automata in this lattice. To restrict Lat$_R$ some more and to refine its top element, one may construct the following table from I$_+$ (resp. for I$_-$ separately). For the Example 2 of the section 7, we obtain (for I$_+$) :

Predecessors of α	$\alpha \in \Sigma$	**Successors of α**
ϵ, a	a	a,b
a, b	b	b, $

The table denotes only the constraint that the membres of $[\alpha]$, $\alpha \in \Sigma$ can preceed/follow some $\beta \in \Sigma$. In the above table, elements of {a,b} can preceede b while elements of {b, \$} can follow b.

Clearly, the table reflects an automaton whose language L_{T+} such that $L_+ \subseteq L_{T+}$. Note that there are recursive parts on a and b and $L_{T+}=a^+b^+$ (there is no a following b). However, considering I_+ (or when only I_+ is available), the table sets an upper bound in Lat_R and the automaton A (to be generated) must be such that $L(A) \subseteq L_{T+}$.

For the Example-2, the table of I_- gives the same content (with F_-). Hence, in this case, $L_- \cap L_+ \neq \emptyset$.

Even though the table may not be of great interest in some cases when $L_- \cap L_+ \neq \emptyset$, it gives some indications on the bounds of Lat_R and hence on the language of the final automaton, specially if I_- enriches L_+.

However, this information, although sometimes not much useful, can avoid some fancy construction in the final automaton. The question of whether I_- may enrich L_+ is crucial in the syntactical learning theory since in the learning process, lessons from the fails can give indications on the successful situations. In the algebraic specification given in this paper, L_+ may contain words that are not represented in I_+. Hence, we may be in a situation where there is no $\hat{a} \in A_{G_I}$ such that $\omega = f(\hat{a})$, $\omega \in L(A)$. In this case, f is not surjective and hence, $\bar{f}$ is *not* an isomorphism.

In the proposed system, the user can activate (or not) the language enrichment function in order to observe experiences on the languages specified by I_+ and I_-. If I_- can enrich L_+ (and vice versa), then L_+ (resp L_-) is "wider", i.e. closer to the top element of Lat_R and hence the final automaton will not be bounded by the tables we construct.

9 Conclusion

In this paper, a new constraint satisfaction for GI has been presented which is implemented by an operational constraint logic program that outputs the final DFA. The algebraic specification allows to show that the homomorphism f exists and we gave an implementation of it by the *Congruence* predicate which produces a set of constraints. If this set is satisfiable, then we choose a solution with the fewest number of states. The lattice of the search can be delimited by the construction of the table of successors and predecessors of any symbol in Σ.

Efficient works on Grammatical Inference deal with positives and negative examples. When only positive examples are available (which describe the characteristic cases), researches concern rather the Structured Documents field and have led to several document standards like ODA and SGML. But in the algebraic and constraint satisfaction frameworks of the Grammatical Inference, the logical aspects for the direct grammar extraction have, as well as known, not yet been investigated.

This work is developed inside a paper document processing project where GI results are used to classify and then translate documents into machine readable form. The generated grammar (under the form of a logic grammar [1]) is augmented to handle some attributes of the logical structure of paper documents such as typographic attributes. We also study also the possibilities offered by *attributed CLP* in handling *ad hoc* unification. Other applications dealing with more general multimedia contents (video in particular) are under the study. The code in GNU-Prolog of the realisation is available from the author.

References

1. A. Colmerauer. "Metamorphosis grammars". In : Natural Communication with Computer. Berlin : Springer-Verlag 1977, LNCS 63, pp. 133-189.
2. P. Van Hentenryck. "Constraint Satisfaction in Logic Programming". MIT Press. Ehud Shapiro Ed. 224 p. 1989.
3. E. Tsang. "Foundations of Constraint Satisfaction". Computation in Cognitive Science. Academic Press 1993.
4. D. Diaz. "Gnu Prolog Reference Manual" INRIA- 2002, http://www.inria.fr/inria/
5. C. de la Higuera. "Current Trends in Grammatical Inference". EURISE, Universit de Saint Etienne. France- 2001.
6. E.M. Gold. "Language identification in the limit". Inf. and Control, 10(5)- 1967.
7. E.M. Gold. "Complexity of automaton identification from given data"". Information and Control, 37- 1978.
8. P. Dupont, L. Miclet & E. Vidal. "What is the search space of Regular Inference?". ICGI'94, Grammatical Inference and Applications. Springer-Verlag-94.
9. P. Dupont. "Regular Grammatical Inference from Positive and Negative Samples by Genetic Search : the GIG method". ICGI'94, Grammatical Inference and Applications. Springer-Verlag-94, LNCS 862.
10. F. Coste, J. Nicols : "Regular Inference as a graph coloring Problem". ICML'97. 1997.
11. J. A. Goguen, J.W. Tatcher, E.G. Wagner, J.B. Wright. "Initial Algebra Semantics and Continuous Algebra". JACM 24(1). 1977.
12. H. Ehrig, B. Mahr. " Fundamentals of Algebraic Specification". Vol-1 & 2. Springer-Verlag1985.
13. J. E. Hopcroft, J.D. Ullmann. "Formal Languages and their Relation to Automata". Addison-Wesley 1969.
14. "Grammatical Inference : Algorithms and Applications", 4th Int. Col. ICGI 1998, Springer-Verlag LNCS. 1433.
15. "Grammatical Inference : Algorithms and Applications", 5th Int. Col. ICGI 2000, Lisbon, Portugal, Springer-Verlag LNCS. 1891.
16. "Grammatical Inference : Algorithms and Applications", 6th Int. Col. ICGI 2002, Springer-Verlag ISBN 3-540-44239-1.
17. SSPR: International Workshop on Advances in Structural and Syntactical Pattern Recognition, LNCS, 2002.

An Evaluation of Alternative Feature Selection Strategies and Ensemble Techniques for Classifying Music

Marco Grimaldi[1], Pádraig Cunningham[1], Anil Kokaram[2]

[1]Computer Science Department, Trinity College Dublin, Ireland
[2]Electronic Engineering Department, Trinity College Dublin, Ireland
{Marco.Grimaldi, Padraig.Cunningham, Anil.Kokaram}@tcd.ie

Abstract. Automatic classification of music files is a key problem in multimedia information retrieval. In this paper we present a solution to this problem that addresses the issues of feature extraction, feature selection and design of classifier. We outline a process for feature extraction based on the discrete wavelet packet transform and we evaluate a variety of wrapper-based feature subset selection strategies that use feature ranking based on *information gain*, *gain ratio* and *principle components analysis.* We evaluate four alternative classifiers; simple *k*-nearest neighbour and o*ne-against-all*, *round-robin* and *feature-subspace* based ensembles of nearest neighbour classifiers. The best classification accuracy is achieved by the feature subspace-based ensemble with the round-robin ensemble also showing considerable promise.

1 Introduction

A key issue in multimedia information retrieval is the need to annotate assets with semantic descriptors that will facilitate retrieval [1]. An example of this is the need to annotate music files with descriptors such as genre. Such a characterization becomes indispensable in scenarios where enhanced browsing systems [2] allow users to inspect and select items from a huge database. One way to automate this process is to label a subset of assets by hand and train a classifier to automatically label the remainder. This is a very challenging machine learning problem because it is a multi-class problem with unresolved questions about how to represent the music files for classification.

In this paper we present a process based on the discrete wavelet packet transform (DWPT) [4] that allows us to represent music files as a set of 143 features. We evaluate a variety of feature selection techniques to reduce this set to a manageable size. We also evaluate four different nearest neighbour classifier techniques:

- Simple *k*-Nearest Neighbour
- One-Against -All Ensemble
- Round-Robin Ensemble
- Feature-Subspace-based Ensemble

We focus on nearest neighbour techniques because of their ease of interpretability, as we will present in section 3. An important objective of this research is to gain some insight into what measurable features predict users tastes in music [2].

When evaluated on a five-class problem with a data-set of 200 music files, we find that the best classification accuracy (84%) is achieved by the feature subspace-based ensemble with the round-robin ensemble also showing considerable promise. While similar music classification tasks have been tackled by other researchers [1, 3, 5] it is difficult to compare results because of the unavailability of benchmark datasets. This will continue to be a problem due to the copyright issues associated with sharing music files.

The paper proceeds with an overview of the music classification problem and a very brief description of the wavelet based feature extraction process in the next section. The different ensemble-based classifiers that are evaluated are described in section 3 and the feature selection process is described in section 4. The details of the evaluation are presented in section 5.

2 The Music Classification Problem

Music information retrieval (MIR), as a research field, has two main branches: symbolic MIR and audio MIR. A symbolic representation of music such as MIDI describes items in a similar way to a musical score. Attack, duration, volume, velocity and instrument type of every single note are available information. Therefore, it is possible to easily access statistical measures such as tempo and mean key for each music item. Moreover, it is possible to attach to each item high-level descriptors such as instrument kind and number. On the other hand, audio MIR deals with real world signals and any features need to be extracted through signal analysis. In fact, extracting a symbolic representation from an arbitrary audio signal (polyphonic transcription) is an open research problem, solved only for simple examples. However, recent research shows that it is possible to apply signal processing techniques to extract features from audio files [1, 3] and derive reasonably sensible classification by genre.

In this work we apply a wavelet packed decomposition to the audio signal in order to decompose the signal spectrogram at two different resolutions; one suitable for frequency-feature extraction, one for time-feature extraction.

2.1 Feature Extraction

The discrete wavelet transform (DWT) is a well-known signal analysis methodology able to approximate a real signal at different scales in time and frequency. Taking into account the non-stationary nature of the input signal, the DWT provides an approximation with excellent time and frequency resolution [4]. The discrete wavelet packet transform (DWPT) [4] is a variant of the DWT. It is achieved by recursively convolving the input signal with a pair of *quadrature* mirror filters: *g (low pass)* and *h (high pass)*. Unlike the DWT that recursively decomposes only the low-pass sub-band, the WPDT decomposes both bands at each level. This procedure defines a grid of Heisenberg Boxes [4] corresponding to musical notes and octaves. Our analysis demonstrates that 9 levels of decomposition are necessary to build a spectrogram suitable for time-feature extraction [6].

Time-features are extracted from the beat histogram. The beat-histogram represents the most intense periodicities found in the signal [5]. The time-features we take into account are: the intensity, the position and the width of the 20 most intensive peaks. The position of a peak is the frequency of a *dominant* beat, the intensity refers to the number of times a beat frequency is found in the song and the width corresponds to the accuracy in the extraction procedure. Additional time-features are: the total number of peaks present in the histogram, its max and mean energy and the length in seconds of the song. A total of 64 time-features are extracted.

Frequency-features are extracted from the spectrum obtained by applying 16 levels of decomposition [6]. Dividing the frequency axis in intervals matching musical octaves, it is possible to characterize the spectrum in a relatively simple way. For every single frequency interval, we calculate the intensity and position of the first 3 most intensive peaks. We record the total number of peaks in each interval, the max and mean energy of the spectrogram as well – 79 frequency-features in total. The total number of features extracted for each song is 143.

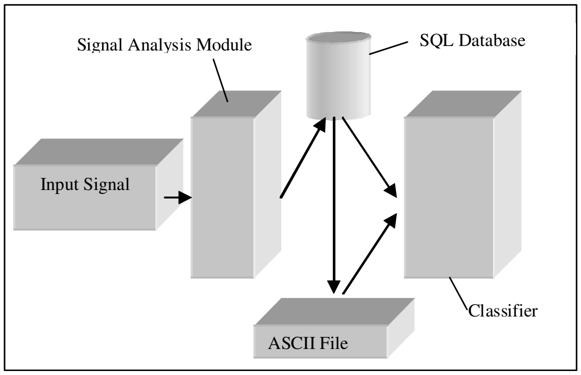


Fig. 1. System Architecture

Characterizing each item with 143 features has advantages and disadvantages. From a knowledge acquisition point of view it is useful to describe an item with the maximum amount of information we can obtain. This is important, because an *a priori* domain description is not available and every feature is potentially useful for classification. Moreover, the music genre classification problem has an implicit difficulty: music genres are not easily definable in terms of low or high level features. On the other hand, dealing with a high dimension feature space brings it own set of problems – perhaps the most important being the increased risk of overfitting. In Section 3, 4 and 5 we show how it is possible to overcome some of these problems.

2.2 Overall System Architecture

The system we have developed has two main units; the first one responsible for signal analysis, the second one of classification. Each audio file is decomposed through the wavelet packet decomposition software and the features are stored into a SQL database.

The classification module can connect to the database to retrieve the item characterization or load the whole dataset as ASCII file. Figure 1 shows the overall system architecture. The module responsible for classification has been designed so that it is possible to choose between different kinds of *k*-NN based predictors (Section 3) and different feature ranking techniques (Section 4).

3 k-NN Based Classifiers

k-NN classifiers are instance-based algorithms taking a conceptually straightforward approach to approximating real or discrete valued target functions. The learning process consists in simply storing the presented data. All instances correspond to points in an *n*-dimensional space and the nearest neighbors of a given query are defined in terms of the standard Euclidean distance [7]. The probability of a query *q* belonging to a class *c* can be calculated as follows:

$$\begin{aligned} p(c \mid q) &= \frac{\sum_{k \in K} w_k \cdot 1_{(kc=c)}}{\sum_{k \in K} w_k} \\ w_k &= 1/d(k,q) \end{aligned} \qquad \textbf{(1)}$$

K is the set of nearest neighbors, *kc* the class of *k* and *d*(*k,q*) the Euclidean distance of *k* from *q*.

Despite their simplicity, *k*-NN classifiers suffer a serious drawback. The distance between items is calculated based on all the attributes. That implies that any features that are in fact irrelevant for classification have the same impact at relevant features. This sensitivity to noise leads to miss-classification problems and to a degradation in the system accuracy. Such a behavior is well known in the literature and is usually referred to as *curse of dimensionality* [7]. In Section 5, we will show that this problem affects heavily the classification accuracy of the system. In fact, not only noisy features affect the classifier accuracy but correlated features may also cause problems.

However, *k*-NN classifiers used in conjunction with effective feature subset selection techniques are readily interpretable and can provide important insight into a *weak theory* domain. Black-box classifiers (e.g. neural nets) do not offer the same insight.

In order to overcome the problem of the high dimensional feature space it is possible to use different strategies. In the next section we present a set of ensemble methods we applied in order to simplify the decision surface the *k*-NN deals with. This simplification is obtained in the ensemble members by reducing the number of classes used to train the k-NN (section 3.1.1 and 3.1.2) or by reducing the feature space dimensionality (section 3.1.3).

3.1 Ensemble Alternatives

An ensemble of classifiers is a set of classifiers whose individual decisions are combined to classify a new item. The final prediction can be derived by weighted or unweighted voting. Research has shown that ensembles can improve on the accuracy of a single classifier, depending on the quality and the diversity of the ensemble members [8]. Ensembles can be implemented in a variety of different ways. In this work we present a comparison of three different ensemble strategies: *one-against-all* (OAA), *round-robin* (RR) [10] and *feature-sub-space* [9] based ensembles.

OAA and RR strategies are used especially with multi-class problems and both work by performing problem-space decomposition. Each ensemble member is a classifier specializing on a two-class problem. On the other hand, each member of the FSS ensemble covers the whole problem space. Each ensemble member is a *k*-NN classifier trained on the same multi-class problem. The improvement due to the ensemble is attributable to aggregation rather than problem decomposition.

3.1.1 One-Against-All Ensemble

As already mentioned, an OAA ensemble performs problem-space decomposition with each ensemble member trained on a re-labelled version of the same data-set. Each component classifier is trained to distinguishing between one single class and its complement in the class space. Thus the number of members in the ensemble is equal to the number of classes in the problem. The probability of a query q belonging to a class c can be calculated as follows:

$$P(c \mid q) = \arg\max_{m \in M}[p_m(c \mid q)] \tag{2}$$

M is the set of ensemble members and $p_m(c|q)$ is the probability given by ensemble predictor m according to equation (1). The big drawback of the OAA technique is that there are no benefits of aggregation; the classification of a given class depends heavily on the member responsible for that class (Even if that member does get to *specialize* on that class).

3.1.2 Round-Robin Ensemble

A RR ensemble converts a c-class problem into a series of two-class problems by creating one classifier for each pair of classes [10]. New items are classified by submitting them to the $c(c-1)/2$ binary predictors. The final prediction is achieved by majority voting. The probability of a query q belonging to a class c can be calculated as follows:

$$P(c \mid q) = \frac{\sum_{m \in M} p_m(c \mid q) \cdot 1_{(mc=c)}}{\sum_{m \in M} p_m(c \mid q)} \tag{3}$$

M is the set of ensemble members, mc is the class predicted by m and $p_m(c|q)$ is the probability given by ensemble predictor m according to equation (1). Clearly, RR is a problem decomposition technique. However there are some aggregation benefits as each class is focused on by c-1 classifiers.

3.1.3 Feature-Sub-Space Ensemble

Sub-sampling the feature space and training a simple classifier for each sub-space is an alternative methodology for building an ensemble. This strategy differs completely from the OAA and RR approaches. It does not decompose the decision space based on the classification task. Instead, the strength of FSS depends on having a variety of simple classifiers trained on different feature sub-sets sampled form the original space. This approach is very similar to a bagging technique where the ensemble is built using different subsets of the instances in the training data. In this work, each ensemble member is trained on different feature-subsets of predefined dimension. Each feature-subset is drawn randomly with replacement from the original set. The probability of a query q belonging to a class c can be calculated according to equation (3).

4 Feature Selection and Ranking Techniques

It is well known that implementing feature selection improves the accuracy of a classifier. The degree of improvement will depend on many factors; the type of classifier, the effectiveness of the feature selection and the quality of the features. In the case of simple k-NN classifier, the feature selection deletes noisy features and reduces the feature-space dimension. Moreover, for an ensemble of classifiers, the feature selection can promote diversity among the ensemble members and can improve their local specialization. The potential for an ensemble to be more accurate than its constituent members depends on the diversity among its members [8].

In this work we consider two approaches to feature selection. We consider a situation where we select the first n features based on one of the ranking criteria. We also consider a wrapper-like [11] forward sequential search that takes a ranked set of feature as starting point. Since the wrapper approach is essentially a greedy search in the feature space for the best feature mask, a key issue in a forward sequential search is the order in which to test the attributes. It is important to start with the more promising attributes. This is in the spirit of Filter/Wrapper algorithms as discussed by Seban and Nock [14]. In the following, we present the ranking algorithms we applied.

4.1 Information Gain

Given entropy (E) as a measure of the impurity in a collection of items, it is possible to quantify the effectiveness of a feature in classifying the training data [7, 13].

$$\begin{aligned} IG(S,A) &= E(S) - \sum_{v \in V(A)} \frac{|S_v|}{|S|} \cdot E(S_v) \\ E(S) &= \sum_{c \in C} - \frac{|S_c|}{|S|} \cdot \log_2 \frac{|S_c|}{|S|} \end{aligned} \qquad (4)$$

Information gain (IG) measure the expected reduction of entropy caused by partitioning the examples according to attribute A.

In the above equation: S is the item collection, |S| its cardinality; $V(A)$ is the set of all possible values for attribute A; S_v is the subset of S for which A has value v; C is the class collection S_c is the subset of S containing items belonging to class c.

It is possible to extend the discrete equation (4) in order to handle continuous-valued attribute. It is done by searching for candidate thresholds sorting the items according to the continuous feature and identifying adjacent items that differ in their target classification [7]. The IG of feature A is equal to the maximum IG value obtained for the various thresholds.

4.2 Gain Ratio

The information gain measure favors attributes with many values over those with few values [7]. *Gain ratio* (GR) overcomes this problem by introducing an extra term taking into account how the feature splits the data:

$$GR(S,A) = \frac{IG(S,A)}{SI(S,A)}$$
$$SI(S,A) = -\sum_{i=1}^{d} \frac{|S_i|}{|S|} \cdot \log_2 \frac{|S_i|}{|S|} \qquad \textbf{(5)}$$

S_i are d subsets of examples resulting from partitioning S by the d-valued feature A. Since the SI term can be zero in some special cases, we define: $GR(S,A) = IG(S,A)$ if $SI(S,A) = 0$ for feature A. For the most part, this improvement over IG proves significant in the evaluation presented here.

4.3 PCA

Principal component analysis (PCA) is a standard technique used to handle linear dependence among variables. A PCA of a set of m variables generates m new variables (the principal components), $PC_1 \ldots PC_m$. Each component is obtained by linear combination of the original variables [12], that is:

$$PC_i = \sum_{j=1}^{m} b_{i,j} \cdot X_j$$
$$\overrightarrow{PC} = B^T \overrightarrow{X} \qquad \textbf{(6)}$$

Where X_j is the j^{th} original variable, $b_{i,j}$ the linear factor. The coefficients for PC_i are chosen so as to make its variance as large as possible. Mathematically, the variation of the original m variables is expressed by the covariance matrix. The transformation matrix B, containing the $b_{i,j}$ coefficients, corresponds to the covariance eigenvector matrix. Sorting the eigenvectors by their eigenvalues, the resulting principal components will be sorted by variance. In fact, the size of an eigenvalue defines how far a feature vector projected onto the eigenspace will be scaled along the correspondent eigenvector direction. Thus this new feature set is naturally ranked by variance which is useful if variance is a reasonable proxy for predictivness. This PCA approach to feature selection has two drawbacks. The first is that it is based on variance of the features only and does not take the class labels into account. The second is that the new features are not readily interpretable.

5 Evaluation and Discussion

In this section we present an evaluation of the different classification techniques presented previously (Section 3). In Section 5.1, we compare the ranking strategies described in section 4 with regard to the kind of classifier (simple *k*-NN, OAA and RR ensemble). In Section 5.2 we evaluate the feature selection applied to the different ensemble strategies. All the classifiers are trained on the same dataset composed of 200 instances divided in 5 different musical genres; with 40 items in each genre. Each accuracy score is obtained by running a stratified 10 fold cross validation experiment. The musical genres we consider are: classical, jazz, techno, rock and heavy metal. The OAA ensemble has 5 members, the RR 10 and FSS ensemble 100. The fact that the FSS ensemble has so many members might not be considered 'fair' and we return to this issue in the Conclusions. The number of *k* nearest neighbours is 5.

5.1 Ranking the Features

The graph presented in Figure 2 show the increase in accuracy of a simple *k*-NN classifiers as features are added based on the three ranking techniques. Each point on the graphs is obtained by running the classification algorithm considering a pre-defined number of features. I.e. 13 features, means that the classification is accomplished considering the 13 best ranked features with respect to the ranking schema selected.

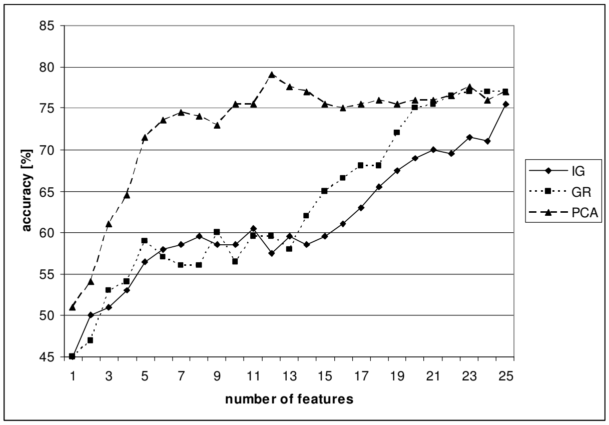


Fig. 2. Comparison of the accuracy score achieved by a simple *k*-NN ranking the features according to information gain, gain ratio and principle component analysis.

Comparing the accuracy behaviour obtained by ranking the features according to IG and GR, it is interesting to note how the system accuracy improves gradually as the number of feature increases. In both cases the classification accuracy doesn't catch up with PCA until a significant number of features are selected (13-20). This kind of behaviour has to be ascribed to correlation among the first 13-20 high ranking features. In fact, the graph shows clearly how PCA improves accuracy by reducing this correlation. Using the first 5 features, the system accuracy increases from 51% to 72%. After a steady state, the accuracy jumps to a value of 79% (12 features).

Figure 3 and 4 shows how the prediction accuracy changes when the ranking techniques are applied to OAA and RR ensembles.

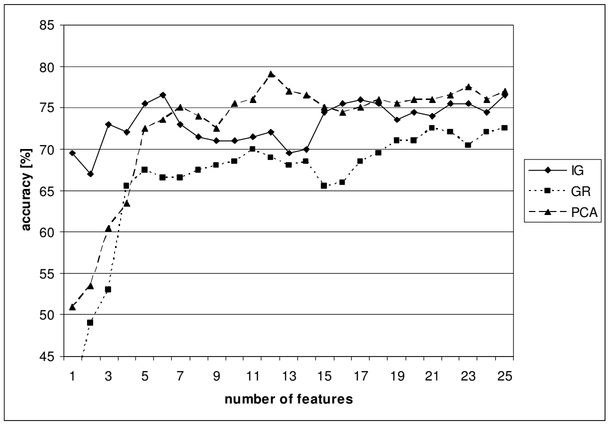


Fig. 3. Accuracy achieved by an OAA ensemble ranking the features using, IG, GR and PCA.

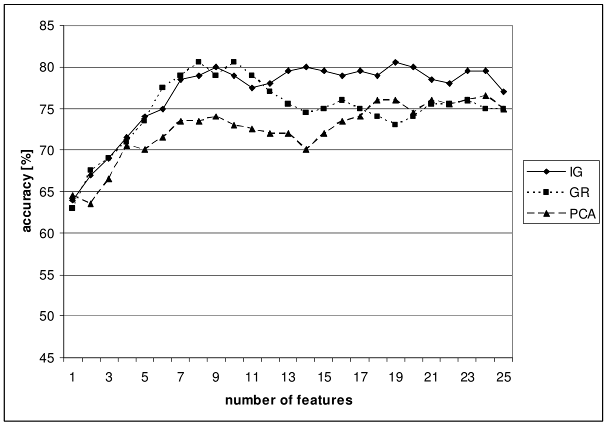


Fig. 4. Accuracy achieved by an RR ensemble ranking the features using, IG, GR and PCA

It is important to point out that the same number of features is selected in each ensemble member. Selecting 10 features implies selecting the first 10 best ranked features in each simple predictor. In this work, the ranking procedure is accomplished

independently in each ensemble member. In this way each simple predictor is locally sensitive to the classification problem it is dealing with.
It is interesting to note that the accuracy obtained by applying PCA matches *exactly* the one presented in figure 2. This fact is due to the lack of diversity in the ensemble: The OAA ensemble is formed by simply re-labelling the instances. Applying PCA in case of simple *k*-NN or OAA ensemble does not change the rank of the features.

Figure 4 shows the results obtained by running the same experiment on a RR ensemble. The RR ensemble seems to be more effective in the classification problem than the other two techniques. Ranking the features according to IG and GR, the ensemble achieves an accuracy of 81%. On the other hand, RR ensemble fails to take advantage of PCA analysis to boost the classification score. This fact is probably due to the poor statistical accuracy of the covariance matrix: the total number of items taken into account decreases from 180 to 72.

5.2 Applying Feature Selection

The forward sequential search algorithm is based on a 10-fold cross validation. In Table 1 we show the system accuracy varying the feature ranking technique and the kind of classifier.

Table 1. Comparison of the prediction accuracies achieved through feature selection, varying the ranking procedure and the kind of classifier.

	IG	GR	PCA
Simple *k*-NN	75%	77%	69%
OAA ensemble	78%	73%	76%
RR ensemble	78%	77%	66%

If we compare these results with those presented in the previous section it is clear that the forward sequential search suffers from over-fitting (simply selecting the top *n* features has better generalisation accuracy than the greedy search). This is due to the small number of example given the large number of features. In this scenario, considering small feature sub-spaces would allow the feature selection to be more effective. In figure 5 we present the results achieved applying the feature selection on a feature sub-space based ensemble. Each point on the graph is obtained running 10 times a stratified 10 fold cross validation. The number of nearest neighbours for each run is 11. The error in the accuracy measure is ± 1% (standard deviation).

Without implementing a feature selection, an ensemble based on sub-spaces of dimension 4 achieves an accuracy of 83%. Applying the forward sequential search, the ensemble accuracy stabilise around 83%, 84% for a large number dimensions (4-10). It is interesting noting that the gain ratio based feature selection gets the best score for different sub-space dimensions: 84%. Given that we are using one small dataset to test the effectiveness of the ensemble methods and the ranking schemas proposed, we cannot claim a generalisation accuracy of 84%. We can say that we would expect an FSS based ensemble with 8 features in each member to produce a very good classifier.

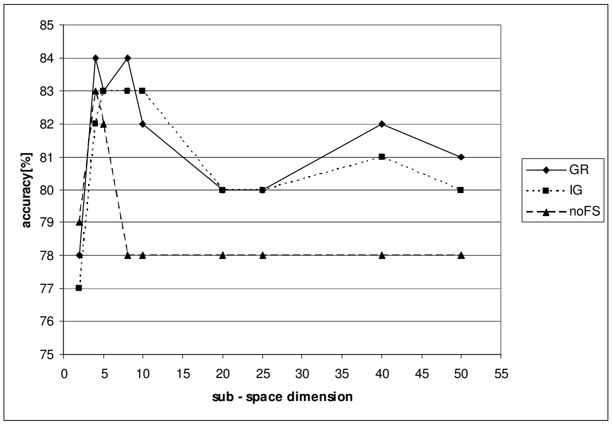


Fig. 5. Accuracy of 3 different feature sub-space ensembles varying the sub-space dimension.

Since a FSS ensemble with 8 features in each classifier is the most promising classifier, we present the confusion matrix in Table 2. This confusion matrix was produced using stratified 10-fold cross validation. The results are even better than the figure of 84% suggests, given that 4% of the error comes from misclassifications between Rock and Heavy Metal. The accuracy for Classical music is 100% although 4 Jazz and 2 Rock tracks are classified as Classical.

Table 2. The confusion matrix for the best classifier developed (e.g. 7 Rock tracks have been classified as Heavy Metal).

Q\A	Jazz	Rock	Techno	Classical	H Metal
Jazz	33	2	1	4	0
Rock	2	27	2	2	7
Techno	2	1	34	0	3
Classical	0	0	0	40	0
H Metal	0	1	5	0	34

6 Conclusion

In this work we have evaluated alternative approaches to the problem of classifying music audio files by genre. Since this is a multi-class problem we have considered the ensemble techniques that are specialised for multi-class – viz, OAA and RR. Because the DWPT process we use for feature extraction produces 143 features we have examined feature ranking and forward sequential search as mechanisms for feature selection. We found that FSS was inclined to overfit the feature selection process but ranking based on GR or IG worked well. In case of simple *k*-NN classifiers, PCA analysis proves to be the most effective feature selection technique, achieving a score of 79%.

The One-against-all ensemble does not appear to be a powerful strategy. The poor diversity between ensemble members is emphasised in figure 3. When PCA is applied, the ensemble technique presents an accuracy behaviour matching exactly that of simple *k*-NN. The Round Robin ensemble scores 81% with both IG and GR, showing to be an effective ensemble technique. However, these classifiers show over-fitting due to the feature selection process. When we apply feature selection to boost the accuracy of the component classifiers, the performance deteriorates. The small number of training examples compared to the number of features is clearly a problem. The best results are achieved with a feature sub-space based ensemble. When we apply a forward sequential search based on the GR ranking, the ensemble scores 84%.

Our evaluation shows that benefits accrue from the problem decomposition that occurs in the RR ensemble but the FSS ensemble wins out because of the aggregation benefits of the large ensemble. The focus of our current work is to bring these two benefits together in a RR ensemble with more than one ensemble member per class pair. This is also identified as a promising avenue of research by Fürnkranz [10].

7 References

[1] Y. Wang, Z. Liu, J.C. Huang, "Multimedia Content Analysis Using Both Audio and Visual Clues", IEEE Signal Processing Magazine, 12-36, November 2000.

[2] C. Hayes, P. Cunningham, P. Clerkin, M. Grimaldi, "Programme-driven music radio", Proceedings of the 15th European Conference on Artificial Intelligence 2002, Lyons France. ECAI'02, F. van Harmelen (Ed.): IOS Press, Amsterdam, 2002

[3] G.Tzanetakis, A.Ermolinskyi, P.Cook: Pitch Histograms in Audio and Symbolic Music Information Retrieval. Proceedings of 3rd International Conference on Music Information Retrieval. ISMIR 2002, Paris, October 2002.

[4] S.G. Mallat, "A Wavelet Tour of Signal Processing", Academic Press 1999.

[5] G. Tzanetakis, G. Essl, P. Cook, "Automatic Musical Genre Classification of Audio Signals", In. Proc. Int. Symposium on Music Information Retrieval (ISMIR), Bloomington, Indiana, 2001.

[6] M. Grimaldi, P. Cunningham, A. Kokaram, "Classifying Music by Genre Using the Wavelet Packet Transform and a Round-Robin Ensemble", Trinity College Dublin, CS Dep., Technical Report, TCD-CS-2002-64, November 2002. (http://www.cs.tcd.ie/publications/tech-reports/tr-index.02.html)

[7] T. M. Mitchell, "Machine Learning", McGraw-Hill, 1997.

[8] G. Zenobi, P. Cunningham, "Using Diversity in Preparing Ensemble of Classifiers Based on Different Subsets to Minimize Generalization Error", 12th European Conference on Machine Learning (ECML 2001), L. De Readt & P. Flach (Ed.), 576-587, Springer Verlag, 2001.

[9] T. G. Dietterich, "Ensemble Methods in Machine Learning", First International Workshop on Multiple Classifier System, Lecture Notes in Computer Science, J. Kittler & F. Roli (Ed.), 1-15. New York: Springer Verlag, 2000.

[10] J. Fürnkranz, "Pairwise Classification as an Ensemble Technique", Proceedings of the 13th European Conference on Machine Learning, pp.97-110, , Springer Verlag, 2002.

[11] R. Kohavi, G.H. John, "The Wrapper Approach", in Feature Selection for Knowledge Discovery and Data Mining, H. Liu & H. Motoda (Ed.), Kluwer, 33-50, 1998.

[12] R.J Harris, "A Primer of Multivariate Statistics", Academic Press, 1975.

[13] J.R.; Quinlan, "C4.5 Programs for Machine Learning", Morgan Kauffman, 1994.

[14] M. Sebban, R, Nock, "A Hybrid Filter/Wrapper Approach of Feature Selection Using Information Theory", Pattern Recognition (35), pp. 835-846, 2002.

Integrated Classification of Audio, Video and Speech using Partitions of Low-Level Features

Edda Leopold[1], Jörg Kindermann[1], Gerhard Paaß[1],
Stephan Volmer[2], René Cavet[2],
Martha Larson[3], Stefan Eickeler[3], and Thorsten Kastner[4]

[1] Fraunhofer Institute for Autonomous intelligent Systems,
Schloss Birlinghoven, 53754 Sankt Augustin, Germany
{Edda.Leopold, Joerg.Kindermann, Gerhard.Paass}@ais.fhg.de
[2] Fraunhofer Institute for Computer Graphics,
Fraunhoferstaße 5, 64283 Darmstadt, Germany
{volmer, rcavet}@igd.fhg.de
[3] Fraunhofer Institute for Media Communication
Schloss Birlinghoven, 53754 Sankt Augustin, Germany
{larson, eickeler}@imk.fhg.de
[4] Fraunhofer Institute for Integrated Circuits
Am Weichselgarten 3, 91058 Erlangen, Germany
ksr@iis.fhg.de

Abstract. Multimodal documents are classified according to the IPTC annotation scheme. To this end usual text classification techniques are adapted to speech, video, and non-speech audio. To represent multimodal documents we apply the bag-of-words approach to speech-, audio-, and video-features. Word analogues are generated for the three modalities: sequences of phonemes or syllables for speech, 'video-words' based on low level color features, and 'audio-words' based on low-level spectral features for non-speech audio. Classification results based on video- or audio-words alone are comparable to those obtained on speech data.

1 Introduction

Content processing of speech, audio, and video data is one of the central issues of recent research in information management. During the last years new methods for the integrated classification of text, audio, video and voice information have been developed. The combination of features from different modalities should lead to an improvment of results. We use low-level features such as color histograms, spectral flatness, and phoneme sequences for the integrated classification of multimodal documents (A/V documents).

Support Vector Machines (SVM) have been applied successfully to text classification tasks [4, 2, 3, 7]. We adapt common SVM text classification techniques to A/V documents which contain speech, video, and non-speech audio data. To represent A/V documents we apply the bag-of-words approach (which is common to text classification). We generate word analogues for the three modalities:

sequences of phonemes or syllables for speech, “video-words” based on low level color features for video, and “audio-words” based on low-level spectral features for general audio.

2 The Motivation of our Approach

2.1 Quantitative Motivation

There is a trade-off between the semantic specificity of signs and their probability of occurrence. Signs with a very specific meaning usually are very rare. Therefore an in-depth analysis of the objects displayed in a picture and the relation between them may lead to sign aggregates which are semantically so specific that they occur very rarely. Such high-level features are likely not to occur in both test set and training set, which makes them useless for supervised classification algorithms. Using low-level features and a partition of the respective feature space for creating audio- and video-signs we are able to control the probability distributions of signs and adjust them properly to subsequent classification procedures.

Furthermore our approach of using low level features is in line with recent linguistic tendencies which prefer shallow parsing techniques rather than an in-depth semantic-syntactic analysis.

2.2 Philosophical Motivation

In his semiotic analysis of pictures Roland Barthes [1] distinguishes between the denoted message and a connoted message of a picture. In his view all imitative arts (drawings, paintings, cinema, theater) comprise two messages: a denoted message, which is the analogon itself, and a connoted message, which is the manner in which the society to a certain extent communicates what it thinks of it.

The denoted message of a picture is an analogical representation (a ’copy’) of what is represented. For instance the denoted message of a picture which shows a person is the person itself. Therefore the denoted message of a picture is a simple agglutination of symbols which is not based on a true system of signs. It can be considered as a message without a code. The connotive code of a picture in contrast results from the historical or cultural experience of a communicating society. The code of the connoted system is constituted by a universal symbolic order and by a stock of stereotypes (schemes, colors, graphisms, gestures. expressions, arrangements of elements). [1]

The rationale behind the use of low-level video features is not to discover the denotated message of a video-artefact (whether it shows for instance a person or a car) but to reveal the implicit code which underlies its connoted message. The elements of the connoted code (i.e. schemes, colors, textures etc.) correspond to what is usually addressed as low-level features in the realm of image processing. Thus as proposed by [6] each element of a partition of feature space can be considered as an (artificial) video-sign of the connoted code whereas the partition itself is the respective vocabulary of video-signs. We extend this idea to non-speech audio features, because determining the denotation of a non-speech audio artefact is usually impossible - not only for technical reasons but simply because it does not exist [9].

3 Feature Extraction

3.1 Speech

The acoustic signal was not separated into speech and non-speech segments. A continuous speech recognition system (CSR) was built using the ISIP (Institute for Signal and Information Processing) public domain speech recognition tool kit from Mississippi State University. It is a typical Hidden-Markov-Model-based system in which the basic acoustic models are phoneme models and consist of three states connected by forward transitions and self-transitions. At each state the probability that a state emitted the given feature vector is modeled by a probability density function composed of a mixture of Gaussians.

Our language model was syllable-based, built by stringing phoneme models together according to a pronunciation lexicon. The advantage of using a syllable-based language model instead of a word-based model is that it leads to reduction of vocabulary-size and results in less out-of-vocabulary errors which makes a domain dependent lexicon unnecessary. This is especially useful when the CSR is applied to a language which is highly productive at the morpho-syntactic level, like German in our case [5]. N-grams were constructed from the recognized syllables and phonemes in order to reach a level of semantic specificity which is comparable to that of words. The use of n-grams also makes it possible to adjust the linguistic units appropriate to the trade-off between semantic specificity and low probability of occurrence, which is especially important when document classes are small. [10]

3.2 Creating Visual and Acoustic Vocabularies

Most results presented in the paper were obtained by using a vocabulary which is drawn from 11 hours of video in october 2002. Vocabularies from the corpus itself and from January 2003 were used only to get an insight in the temporal variation of the visual code. Interestingly comparison of results based on the different vocabularies did not differ very much.

Video The generation of a visual vocabulary is done on a corpus of video data from recorded TV news broadcasts. In our tests, the video set for training had an approximate length of eleven hours. First the video data was split into individual frames. To reduce the huge amount of frames, only one frame per second of video material was selected . This could be done because of the high similarity between the neighboring frames. This reduces the frames count from approx. 500000 to

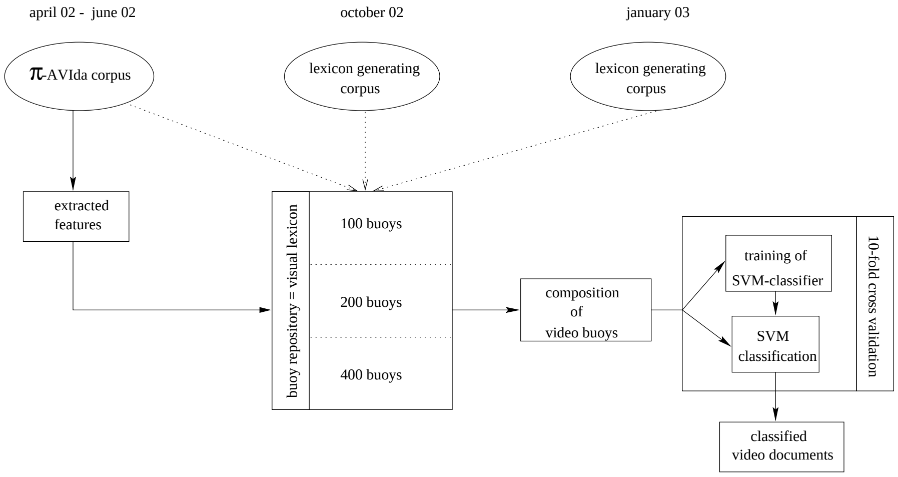

april 02 - june 02
october 02
january 03
π-AVIda corpus
lexicon generating corpus
lexicon generating corpus
extracted features
buoy repository = visual lexicon
100 buoys
200 buoys
400 buoys
composition of video buoys
training of SVM-classifier
SVM classification
10-fold cross validation
classified video documents

Fig. 1. Outline of our experimental design (video only) Most results presented in the paper were obtained by using a vocabulary which was drawn from the october 2002 data. Vocabularies from the corpus itself and from January 2003 were used only for comparison.

17342 frames. After that, the content feature descriptors were extracted on the reduced set of frames and the buoy generation method [11] is applied to each descriptor set.

Three low-level features were extracted from each key-frame: a histogram of 29 colors, a correlogram calculated on the basis of 9 major colors, and first and second moment of the distributions of each the 9 major colors. For each of these features vocabularies of different sizes (100, 200, 400, and 800) of video-buoys were generated. They represent visual vocabularies (or visual lexicons) of different sizes. A larger lexicon size implies more specific video-signs, each covering a smaller semiotic extension on average. Buoy sets were generated also from video data recorded at different time periods: April 2002 to July 2002 (this data set comprises the multimodal corpus described below i.e. the A/V scenes to be classified whereas the other sets were solely used for lexicon generation), October 2002, and January 2003. This was done in order to get an insight into the temporal variation of the visual lexicon.

Audio The low-level audio features that we considered were Audio-Spectrum-Flatness and Audio-Spectrum-Envelope as described in MPEG-7-Audio. Audio-Spectrum-Flatness was measured for 16 frequency bands ranging from 250 Hz to 4 kHz for every audio frame of 30 msec. The Audio-Spectrum-Envelope was calculated for 16 frequency bands ranging from 250 Hz to 4 kHz plus additional bands for the low-frequency (below 250 Hz) and high-frequency (above 4 kHz) signals. For both features a vocabulary of 1000 video-buoys was generated. We are in the process of experimenting with different sizes and creation dates of the acoustic vocabulary. These experiments, however, have not yet been completed.

3.3 Mapping A/V Scenes to Audio- and Video-Words

The A/V stream is segmented manually into semantically coherent scenes that belong to one news story. The first step for representing the video scenes is the segmentation of the video stream into coherent units (shots). For each shot a representative picture is selected, the "key frame". The segmentation is done by algorithms monitoring the change of image over time. Two adjacent frames are compared and their difference is calculated. The differences are summed, and when the sum exceeds a given threshold a shot-boundary is detected, and the key frame of the shot is calculated. For each shot the three low-level features extracted from its key frame. Each of the three visual features is mapped to the nearest video-buoy in the respective visual vocabulary. The three buoy IDs are concatenated to form a *video-word*, which represents the shot.

Both acoustic feature vectors are mapped to the nearest audio-buoy of its respective vocabulary. This way every audio-frame of 30ms is represented by two audio-buoys, which are combined to form an *audio-word.*

Repeated sequences of *identical* video or audio-words are reduced to a single video- or audio-word. Resulting video- or audio-words are combined to form sequences (n-grams) up to a length of $n \leq 5$. Further processing follows the usual bag-of-words representation which is commonly applied for text classification: For each scene (and for each of the three modalities: video, audio, and speech) a type-frequency vector is generated which contains the number of occurrences for each n-gram. These type-frequency vectors are concatenated and span a product space for the integrated multi-modal classification.

The major difference between audio and video-processing is that audio-words (for the time being) were generated for each audio-frame (30 msec) whereas video-words were created for every shot, which may last up to 3 sec. The poor results for audio classification reported below show that this technique has to be improved and that audio-words have to be created for temporally larger units. Respective experiments are in progress but have not been finished yet.

4 The Multimodal Corpus

The corpus consists of 693 multimodal (A/V)-documents. Segmentation into semantically coherent scenes (documents) and semantic annotation according to the categorization scheme of the International Press Telecommunications Council (IPTC) (see http://www.iptc.org) was done manually. The data were obtained from two different German news broadcast stations: N24 and n-tv. Document length ranges between 30 sec. and 3 minutes. The material from N24 consists of 353 A/V-documents and covers the period between May 15 and June 13, 2002 (including reports from the World Cup soccer tournament) in Korea and Japan, which can be considered as a semantically unique event, that does not appear in the “vocabulary” obtained from tv recordings of November). The data from n-tv comprises 340 documents and covers the last seven days of April 2002. Table 1 shows the distribution of topic classes in the corpus. For convenience we added two classes “advertisement” and “jingle” to the 17 top level classes of the IPTC-categorization. The number of documents in the classes total more than 693, because some documents were attributed to two or three classes because of the ambiguity of their content. For example A/V-documents on the Israel-Palestine conflict often were categorized as belonging to both “politics” and “conflicts”.

Note that the size of the classes varies considerably; “politics” comprises 200 A/V-documents whereas “religion” contains just 4. We only used the seven categories with more than 45 documents (shown in the right columns of table 1) for classification experiments. This means that all documents of the other (small) categories were always in the set of counter-examples.

Table 1. Size of IPTC-classes in terms of number of documents. Only those classes, which contain more than 45 documents (right columns) were considered

category	docs.	category	docs.
religion	4	labour	49
social issue	6	economy	68
weather	8	conflicts	85
education	10	sports	91
science	13	advertisement	119
leisure	15	justice	120
environmental issue	17	politics	200
health	19		
culture	22		
jingles	22		
disaster	38		
human interest	40		

5 The Classification Procedure

5.1 Preprocessing

Each video scene d_i is represented by its type-frequency vector

$$\mathbf{f}_i = \big(r_1 \cdot f(w_1, d_i), \ldots,\ r_n \cdot f(w_n, d_i)\big) \tag{1}$$

where r_j is an importance weight as described below, w_j is the j-th n-gram (or the j-th type generated by the visual vocabulary), and $f(w_k, d_i)$ indicates how often w_j occurs in the video scene d_i. Type-frequency vectors are normalized to unit length with respect to L_1. In subsequent tables the use of type-frequencies is indicated by "rel". The vector of logarithmic type-frequencies of a video scene d_i is defined as

$$\mathbf{l}_i = \Big(r_1 \log\big(1 + f(w_1, d_i)\big), \ldots, r_n \log\big(1 + f(w_n, d_i)\big)\Big) \tag{2}$$

Logarithmic frequencies are normalized to unit length with respect to L_2. Other combinations of norm and frequency transformation were omitted because they appeared to yield worse results. In tables below the use of logarithmic type-frequencies is indicated by "log".

As there is a large number of possible n-grams in the scenes, a statistical test is used to eliminate unimportant ones. First it is required that each type must occur at least twice in the corpus. In addition the hypothesis that there is a statistical relation between the document class under consideration and the occurrence of a type w is investigated by a χ^2-statistic. A type w is rejected when its χ^2 statistic is below a threshold θ. The values of θ used in the experiments are $\theta = 0.001$ and $\theta = 1$.

Importance weights like the well-known inverse document frequency are used often used in text classification in order to quantify how specific a given type is to the documents of a collection. Here another importance weight, namely redundancy, is used. In information theory the usual definition of redundancy is maximum entropy ($\log N$) minus actual entropy. So redundancy is calculated as follows: consider the empirical distribution of a term over the documents in the collection and define the importance weight of type w_k by

$$r_k \quad = \quad \log N + \sum_{i=1}^{N} \frac{f(w_k, d_i)}{f(w_k)} \log \frac{f(w_k, d_i)}{f(w_k)}, \tag{3}$$

where $f(w_k, d_i)$ is the frequency of occurrence of term w_k in document t_i and N is the number of documents in the collection. The advantage of redundancy over inverse document frequency is that it does not simply count the documents that a type occurs in, but takes into account the frequencies of occurrence in each of the documents. Since it was observed in previous work [7] that redundancy is more effective than inverse document frequency, two experimental setting are considered in this paper: term frequencies $f(w_k, d_i)$ are multiplied by r_k as defined in equation 3 (denoted by "+" at column "r" in subsequent tables); or term frequencies are left as they are: $r_k \equiv 1$ (denoted by "–").

5.2 Classification

We use a soft margin Support Vector Machine (SVM) with assymetric classification cost in a 1-vs-n setting. I.e. for each class a SVM was trained which separates this class against all other classes in the corpus. The cost factor by which the training errors on positive examples outweight errors on negative examples is set to $j = 2\frac{\#neg}{\#pos}$, where $\#pos$ and $\#neg$ are the number of positive and negative examples respectively. This means that the weighting of false positive training errors is larger for smaller classes and in the case $\#neg = \#pos$

positive examples on the wrong side of the margin are weighted twice as much as negative examples. The trade off between training error and margin was set to $C = \sum_{i=1}^{\ell} \|x_i\|^{-1}$ which is the default in the SVM implementation that we used.

It is well known that the choice of the kernel function is crucial to the efficiency of support vector machines. Therefore the data transformations described above were combined with the following different *homogeneous* kernel functions:

- Linear kernel (L)
  $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j$
- 2nd and 3rd order polynomial kernel (P(d))
  $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j)^d \qquad d = 2, 3$
- Gaussian rbf-kernel (R(γ))
  $K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|} \qquad \gamma = 0.2, 1, 5$
- Sigmoidal kernel (S)
  $K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\mathbf{x}_i \cdot \mathbf{x}_j)$

In some of the experiments (see table 8) these *homogeneous* kernel functions were combined to form *composite* kernels, which use different kernel functions for each modality (for example L for speech, R(1) for video and P(3) for audio). Formally a composite kernel is defined as follows: Let the input space consist of L_s speech attributes, L_v video attributes, and L_a audio attributes, which are ordered in such a way, that dimension 1 to L_s correspond to speech attributes, dimensions $L_s + 1$ to $L_s + L_v$ correspond to video attributes, and dimensions $L_s + L_v + 1$ to $L_s + L_v + L_a$ correspond to audio attributes. Let $\pi_l^k(\cdot)$ be the projection from the input space to its subspace spanned by dimensions k to l. A composite kernel that uses kernel K_1 for speech, K_2 for video and K_3 for audio is defined as

$$\begin{aligned} K_{K_1,K_2,K_3}(\mathbf{x}_i, \mathbf{x}_j) = {} & K_1\left(\pi_{L_s}^1(\mathbf{x}_i), \pi_{L_s}^1(\mathbf{x}_j)\right) \\ & + K_2\left(\pi_{L_s+L_v}^{L_s+1}(\mathbf{x}_i), \pi_{L_s+L_v}^{L_s+1}(\mathbf{x}_j)\right) \\ & + K_3\left(\pi_{L_s+L_v+L_a}^{L_s+L_v+1}(\mathbf{x}_i), \pi_{L_s+L_v+L_a}^{L_s+L_v+1}(\mathbf{x}_j)\right) \end{aligned}$$

The idea behind the construction of composite kernels is that the different semiotic and cognitive conditions for speech, video and audio imply different geometries in the respective factor spaces. I.e. we treat audio, video, and speech differently although we represent them in the same imput space. A kernel is called a homogeneous kernel if $K_1 = K_2 = K_3$.

6 Results

The following tables show the classification results on the basis of the different modalities. A "+" in the column "r." indicates that the importance-weight redundancy is used, and "-" indicates that no importance weight is used. The

Table 2. Results of the classification on the basis of phoneme sequences.

category	n	r.	θ	tra.	kernel	F-score
justice	3	+	1.00	rel	R(5)	67.3
economy	3	-	1.00	log	S	60.2
labour	3	+	1.00	rel	P(2)	83.6
politics	4	+	1.00	rel	R(5)	74.7
sport	3	-	1.00	log	S	84.8
conflicts	4	+	1.00	rel	R(5)	72.2
advertis.	3	-	1.00	log	P(2)	88.9

values of the significance threshold θ used in our experiments are $\theta = 0.01$ and $\theta = 1$. The column "tra." indicates the frequency transformation that was used, "log" stands for logarithmic frequencies with L_2-normalization and "rel" means relative frequencies (i. e. frequencies with L_1-normalization). The next column "kernel" indicates the kernel function: L is the linear kernel, S is a sigmoidal kernel, and $P(d)$ and $R(\gamma)$ denote the polynomial kernel and the rbf-kernel respectively. The last column shows the classification result in terms of the $F - score$, which is calculated as

$$F = \frac{2}{\frac{1}{prec} + \frac{1}{rec}},$$

where rec and $prec$ are the usual definitions of the recall and precision [8]. For the classification a 1-to-n scheme was used, i.e. each class was classified against all other documents. All classification results presented in this section were obtained by tenfold crossvalidation, where the lexicon is held constant. This makes the results statistically reliable. Crossvalidation involving lexicon generation is unnecessary because the data set used for lexicon generation is different from the multimedia corpus.

6.1 Classification Based on Speech

The results on speech-based classification for the optimal combinations of parameters are presented in in tables 2 and 3. From the speech recognizer output phoneme-n-grams and syllable-n-grams were constructed for $N = 1$ to $n = 5$. Table 2 shows that classification was based on sequences of up to 4 phonemes. This exceeds the average syllable length in German, which is about 2.9 phonemes. For phonemes there is no clear tendency in the choice of kernels, in contrast to the case of syllable-sequences, where most classes were best classified with rbf-kernels (cf. table 3). Syllable based classification is slightly worse thanb the one based on phonemes. This is in line with earlier experiments [10].

6.2 Classification Based on Video

Table 4 shows results based on a set of 100 video buoys, table 5 those on 400 video buoys. In the case of a visual lexicon of 100 video buoys the units used

Table 3. Results of the classification on the basis of syllable sequences.

category	n	r.	θ	tra.	kernel	F-score
justice	1	+	1.00	rel	R(1)	65.0
economy	2	+	1.00	rel	P(2)	59.3
labour	1	+	1.00	rel	R(1)	85.3
politics	2	+	1.00	rel	R(0.2)	74.7
sport	1	+	1.00	rel	R(5)	80.3
conflicts	2	+	1.00	rel	R(0.2)	73.5
advertis.	1	-	1.00	log	R(0.2)	85.0

for the classification are n-grams with a size varying from $n = 1$ to $n = 3$ (note that such a unit may last up to 5 seconds). This means that these units are built from one to three video-shots. Those categories that are classified on the basis of shot-unigrams show relatively poor performance. We therefore suppose that we have detected regularities in the succession of video-units, which reveal a kind of temporal (as opposed to spatial) video-syntax. Linear kernels never perform best for any category. Rbf-kernels seem to be the most appropriate for classification on the basis of video-words when a small set of video buoys is considered.

With more buoys to choose from, the performance increases significantly with the exception of the category 'labour'. The n-gram degree decreases, and there is a large variety of kernel functions. We attribute this to the fact that the semantic specificity of n-grams increases with n. As units from a larger vocabulary are on average semantically more specific than units from a smaller one, the specificity of video-words obtained from the larger vocabulary is compensated by a decrease of n-gram degree.

Table 4. Classification results based on a visual vocabulary of 100 video buoys

category	n	**r.**	θ	**tra.**	**kernel**	F
justice	2	–	1.00	rel	R(1.0)	49.796
economy	3	+	1.00	rel	R(5)	24.490
labour	1	–	0.01	rel	S	33.333
politics	3	–	1.00	rel	S	43.928
sport	3	+	1.00	rel	R(0.2)	46.739
conflicts	1	–	0.01	rel	R(1.0)	35.000
ads	2	+	0.01	rel	R(5.0)	84.581

The effect of date of lexicon creation One may argue that using video buoys which were generated after the acquisition of the test corpus may be flawed because typical pictures cannot be present in video material obtained at

Table 5. Classification results based on a visual vocabulary of 400 video buoys

category	n	**r.**	θ	**tra.**	**kernel**	F
justice	1	−	1.00	log	S	51.6
economy	1	+	1.00	log	L	39.3
labour	1	−	0.01	log	S	16.1
politics	2	−	1.00	log	P(3)	57.4
sport	2	+	1.00	log	S	49.7
conflicts	2	−	0.01	log	R(0.2)	44.7
ads	2	+	0.01	rel	R(5.0)	90.0

a later interval of time. However our principal assumption was that the video buoys reveal a kind of implicit code, which is known to the individuals of a given society. The assumption of such a code implies that it is shared by the members of the society and functions as a means to convey (non-linguistic) information. To fulfil this communicative function the code may not vary too quickly. As can be seen in figure 3, experimental results with visual lexicons created at different times (summer 2002 and January 2003) did not show a consistent change in performance and support the assumption of independence from the date of lexicon acquisition.

From figure 3 one can see that an effect of a steady evolution of the visual semiotic system is reflected in the results of the classification. Categories justice and sports and to a lesser extent politics show a sharp decline of performance when the lexicon was drawn from the October material instead of the corpus itself. This can attributed to the fact that there were salient news in these categories at the time when the corpus was sampled, namely the socker world championship (sports) and a massacre at a German high school (justice).

The results for the categories economy and conflicts are nearly independent from the creation date of the visual lexicon. These categories are communicated by visual signs which seem to be temporarily invariant as far as they can be described by color based low-level features.

The category labour again plays a special role. On the video buoys obtained from corpus itself this category shows reasonable classification performance. However there is a poor temporal generalization. It might be that in the case of labour the artificially generated video-signs do not correspond to those visual signs which are actually used in the communicating society.

6.3 Classification Based on General Audio

Classification on the basis of the audio-words is shown in table 6. The error rates are worse than those of the other two modalities (speech and video). Those categories which show a good performance mostly are classified on the basis of audio-word unigrams. This means that building sequences of audio-words in general does not improve the performance. Keeping in mind that audio-words are generated for every audio-frame of 30 msec., this suggests that there are only

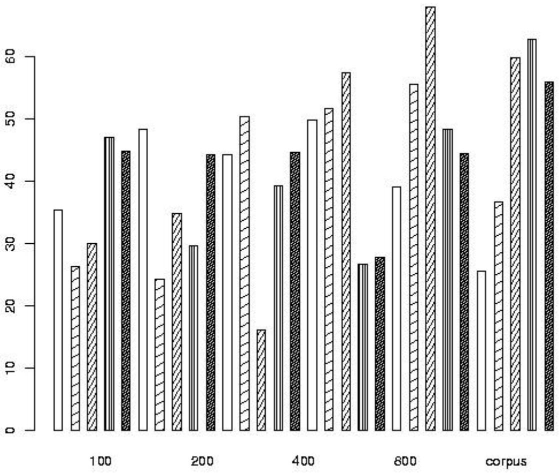


Fig. 2. Classification performance vs. vocabulary size. All vocabularies except corpus where obtained from October 2002. One can see that different classes show different behaviour when the size of vocabulary is changed. Note that the visual vocabulary which was obtained from the corpus itself (labeled as "corpus") does not yield outstanding classification results compared to the other

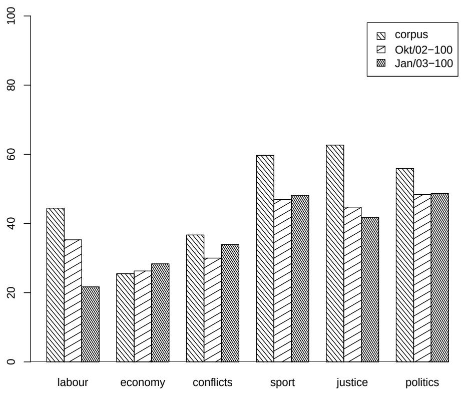


Fig. 3. Classification performance vs. date of vocabulary acquisition

little regularities in the temporal combination of audio-units at this timescale. Performance may improve when audio-words are generated for larger time intervals. Polynomial-kernels seem to be most appropriate for classification on the basis of audio-words.

Table 6. Results of the classification on the basis of sequences of audio-words

category	n	r.	θ	tra.	kernel	F-score
justice	1	-	0.01	rel	R(5)	37.57
economy	4	-	1.00	rel	P(3)	15.13
labour	1	-	1.00	rel	P(3)	13.33
politics	1	-	0.01	rel	R(5)	41.96
sport	1	-	1.00	rel	P(2)	28.57
conflicts	3	+	1.00	rel	P(3)	21.43
advertis.	1	-	0.01	log	S	90.60

We have not yet experimented with different dates of lexicon acquisition and different sizes of acoustic vocabularies. This is ongoing work as well as the utilization of acoustic units spanning a larger interval of time.

6.4 Comparison of Results from Video, General Audio, and Speech

Figure 7 compares the error rates of all modalities directly. For most classes phoneme sequences yield best results. In some of the cases where phoneme sequences are comparatively bad, other modalities improve the picture.

Table 7. Comparison of the results on different modalities

category	phon.	syl.	video	audio	docs.
justice	**67.3**	65.0	49.80	37.57	120
economy	**60.2**	59.3	24.49	15.13	68
labour	83.6	**85.3**	33.33	13.33	49
politics	74.7	**74.7**	43.93	41.96	200
sport	**84.8**	80.3	46.74	28.57	91
conflicts	72.2	**73.5**	35.00	21.43	85
advertis.	88.9	85.0	84.58	**90.60**	119

Classification on the basis of audio-words yields worse results than the other two modalities (speech and video). There is however one big exception: advertisement. The generally superb rates of the category “advertisement” are likely to be caused by the following: The shot duration in commercial spots is generally very short, and therefore it exhibits a temporal visual syntax different from other categories. Furthermore the overall energy in the audio spectrum tends to be considerably higher than normal: advertisements sound more "intense" to the human ear than other broadcasts. Another aspect is that commercials are broadcast repeatedly. This means that in some cases identical spots tend to be present in both test and training data.

Table 8. Results of the classification using composite kernels. The vocabularies were created from 100 video-buoys and 1000 audio-buoys. Speech was represented in terms of syllables

				n-gram			kernel		
cat.	r.	tra.	a	s	v	a	s	v	F
justice	+	rel	1	2	2	S	P(2)	P(2)	50.8
econ.	+	log	1	1	1	P(2)	S	R(1)	45.2
labour	-	log	1	1	1	P(3)	S	R(1)	58.2
politics	+	rel	1	1	1	P(3)	R(1)	R(1)	62.0
sport	+	rel	1	1	1	P(2)	R(1)	R(1)	49.3
confl.	+	rel	1	1	2	P(2)	R(1)	R(1)	45.7
adv.	-	rel	1	1	1	P(2)	R(1)	S	88.6

6.5 Integrated Classification using Composite Kernels

We explored the use of composite SVM kernels which apply different input space geometries for different modalities. Table 8 shows preliminary results. The columns labeled "a", "s", and "v" show the n-gram degrees and kernel parameters for audio, speech, and video sections of the sign frequency vectors. Comparison with table 7 reveals that results are as yet inferior to the best results based on single modalities. The results presented in table 8 have been obtained with a visual inventory of 100 video-buoys.

6.6 Integrated Classification using Homogeneous Kernels

Further experiments were carried out with homogeneous kernels and different sizes of the visual vocabulary combined with speech. It turned out that phoneme-sequences and a visual vocabulary of 400 buoys yield the best results (see table 9). The classes "advertisements" and "sports" show an improvement in accuracy compared with the single modalities of 3 %. For the other classes the F-score for integrated classifcation lies between the respective values for phonemes (table 2) and video (table 5). Interestingly the optimal parameters have changed compared to the single modalities: sigmoidal kernels perform best for all classes and the length of phoneme sequences is reduced. This suggests that representing the type-frequency vectors from different modalities in a common product space may not be the optimal strategy for the application on SVM to the classification of multimodal documents.

Table 9. Results of the classification with sequences of video-words (400 buoys) and syllables

category	n (video)	n (speech)	r.	θ	tra.	kernel	F-score
justice	3	3	-	1.00	log	S	62.7
economy	2	3	+	1.00	log	S	59.2
labour	1	3	-	1.00	log	S	78.0
politics	1	2	+	1.00	log	S	65.3
sport	1	2	+	1.00	log	S	86.6
conflicts	2	2	+	1.00	log	S	71.0
advertis.	1	2	+	1.00	log	S	93.4

7 Conclusion

Audio- and video-words constructed from low-level features provide a good basis for the classification of A/V-documents. The best F-scores on our difficult corpus range between 50% and 90%. Visual vocabularies generated as described in this paper are to a certain extent temporally stable. This allows to create a visual lexicon before the actual video classification is performed. The classification performance depends on the lexicon size. As units from a larger vocabulary are on average semantically more specific than units from a smaller one, the specificity of video-words obtained from the larger vocabulary is compensated by a decrease of n-gram degree. Using sequences of audio-words generated for every single audio-frame of 30 msec. yields poor classification performance (except for commercials). This suggests that there are no regularities in the temporal combination of audio units at this timescale. Performance may improve when audio-words are generated from larger time intervals. Composite kernels which induce different geometries on different modalities do not lead to an improvement of classification. However a proper adjustment of lexicon sizes of the different modalities is crucial to a successful integrated classification of multimodal documents. It is questionable if representation of the different modalities in a joint product space is the best solution to the problem of media integration.

8 Acknowledgment

This study is part of the project Pi-AVIda which is funded by the German ministry for research and technology (BMFT) (proj. nr. 107). We thank the Institute for Communication and Phonetics of the University of Bonn for contributing the BOSSII system and we thank Thorsten Joachims (Cornell University) who provided the SVM-implementation SVM^{light}.

References

1. Barthes, R.: Image, Music, Text. Noonday Press (1988)
2. Dumais, S., Platt, J., Heckerman, D., Sahami, M.: Inductive learning algorithms and representations for text categorization. In: 7th International Conference on Information and Knowledge Management (1998)
3. Drucker, H., Wu, D., Vapnik, V. Support vector machines for spam categorization. In: IEEE Transactions on Neural Networks **10** (1999) 1048–1054
4. Joachims, T.: Text categorization with support vector machines: learning with many relevant features. In: Proceedings of the Tenth European Conference on Machine Learning (ECML 1998) Springer Lecture Notes in Computer Science, Vol. 1398, Springer-Verlag (1998) 137–142
5. Larson, M., Eickeler, S., Paaß, G., Leopold, E., Kindermann, J.: Support Vector Machines for German Spoken Document Classification. In: Proceedings of the International Conference of Spoken Language Processing (ICSLP) vol. 3, (2002) 1989–1992.
6. Leopold, E.: Artificial Semiotics. 10th International Congress of the German Society of Semiotics (DGS) (2002).
7. Leopold, E., Kindermann, J.: Text Categorization with Support Vector Machines. How to Represent Texts in Input Space? Machine Learning, **46** (2002) 423–444
8. Manning, C., Schütze, D.: Foundations of Statistical Natural Language Processing. MIT Press (1999)
9. Nattiez, J.-J.: De la sémiologie à la musique. Université du Québec à Montreal, Montreal (1988)
10. Paaß, G., Leopold, E., Larson, M., Kindermann, J., Eickeler, S.: SVM Classification Using Sequences of Phonemes and Syllables. In: Elomaa T., Mannila H., Toivonen H. (eds.): Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD 2002), Springer (2002) 373–384
11. Volmer, S.: Fast Approximate Nearest-Neighbor Queries in Metric Feature Spaces by Buoy Indexing. In: Proc. 5th International Conference on Visual Information Systems, Hsinchu Taiwan (2002) 36–49

Knowledge Representation of Low Level Features for Semantic Video Analysis

Andres Dorado and Ebroul Izquierdo

Queen Mary, University of London,
Electronic Engineering Department,
Mile End Road, London E1 4NS, UK
{andres.dorado, ebroul.izquierdo}@elec.qmul.ac.uk

Abstract. An approach towards automatic knowledge representation for semantic-based video analysis is described. The approach consists of two stages: learning and generalization. In the learning stage, fuzzy sets are used to map low level features into a set of user-specified keywords. Frequent patterns representing associations between low level features and semantic concepts are extracted applying association rule mining. These patterns are used to build an inference rule base. In the generalization stage, the inference rule base is used to automatically associate appropriate keywords to video clips. Experimental results show that this approach is suitable to support analysis of video content.

1 Introduction

Important advances in content-based retrieval (CBR) systems have been done in the last decade ([1]-[5]). However, with the amount of available video data, users are demanding more robust and user-friendly applications for automatic video content analysis and efficient browsing.

To achieve this vision important functionalities are required: system tools facilitating the manual annotation and editing task, algorithms and user interfaces to speed up the content and context based annotation, analysis, and editing with the "human-in-the-loop", and shift the model from camera-centered to scene-centered [6].

In addition, CBR systems must incorporate reasoning and learning capabilities to establish an intelligent architecture, which enable them to produce effective and accurate results according to the requests of the end-users.

Advanced CBR systems combine low level features with high level concepts to generate interpretations of video content. Interpretation in the video understanding context is mainly focused on "what is happening in a scene" instead of "what is in the scene" [7].

These interpretations also called semantic concepts can be represented by symbols such as keywords and icons. The association of these symbols to video segments is known as *video annotation*.

To achieve efficient results in video annotation, contributions from fields such as cognitive psychology, artificial intelligence, and semiotics are required[8].

Besides, video annotation as well as image annotation requires a framework. Perceptual-based video annotation can take into account the following frameworks summarized by Salway et al: *physical*, visual phenomena become perceptually meaningful; *diegetic*, the 4-D spatio-temporal world posited by the film; *connotative*, metaphorical analogical and associative meanings; and *subtextual*, specialized meanings of symbols and signifiers [9].

Video annotation is not limited to attach text to frames or segments. The complexity of this process can be illustrated with the purpose of any film video annotation: "to discover and represent the film content in all its aspects: story unwrapping, the mastery of the actors, director, and cameraman, editing effects, etc. Thus, the film and its context will become more accessible and clearer to the user" [10].

As it is described in Section 2, annotations can be useful for video analysis. One example is the visual query language proposed by Hibino and Rundensteiner, which temporally links annotations to a video segment and spatially links them to a position on the video [11]. The linked annotations contain descriptive objects with semantic information about the events in the video segment.

In this paper, *video analysis* refers to semantic interpretations of the video content. This analysis is carried out vertically and horizontally. Vertically, because it goes from low level to high level features. Horizontally, because of some results can be achieved with a multimodal model only.

An approach for automatic knowledge representation is presented in Section 3. The approach consists of two stages: learning and generalization.

In the learning stage, low-level features are automatically extracted. Using fuzzy sets, low level features are mapped into a set of user-specified keywords. These fuzzy sets are automatically generated from the data applying fuzzy clustering.

Frequent patterns representing associations between low level features and semantic concepts are extracted applying association rule mining. A strategy for propagating these concepts based on visual properties is described in [12].

These patterns are used to build an inference rule base, which is based on the Fuzzy Associative Memory (FAM) model. This FAM contains rules of the form: "IF *condition* THEN *action*". FAM-based inference system can be used for analysis of video both in low- and high- level. In [13] a FAM-based fuzzy inference for detecting shot transitions is presented. The inference rule base completes the knowledge representation.

In the generalization stage, knowledge representation is used to automatically associate appropriate semantic concepts for annotating video clips. The inference base on the FAM model can be suitable to support analysis of video content.

The experiments reported in Section 4 were applied on visual properties. However, the approach supports other kind of data, overcoming the so called "silent-video" syndrome[14].

2 Video Annotation

The term *annotation* refers to the use of auxiliary symbols that are used to modify the interpretation of other symbols. These annotation symbols typically do not have the same kind of meaning as the symbols that they annotate [15].

Intille and Bobick define *video annotation* as "the task of generating descriptions of video sequences that can be used for indexing, retrieval, and summarization"[16]. Such descriptions are aimed to associate semantic meaning with video segments to improve the content-based retrieval. For their generation, low level data is processed and linked to high level interpretations. Therefore, video annotation plays an important role in video analysis.

Typically, video annotations are represented by abstract elements known as *video annotation objects*. These objects are temporally and spatially linked to video segments. Each video annotation object contains descriptors characterizing the video content. These objects can be extended to more complex structures such as multimedia objects [17].

Video annotation requires uniform models for representing its content and facilitating the use to several users. One way for getting durable and stable representations is using icon-based languages such as the visual query language proposed by Davis [18]. VIRON is an example of efficient mechanisms to support sharing and reusing annotations among users [19].

An approach that supports video annotation is presented in the next section. It generates and uses a knowledge representation for establishing a bridge between low level features and their interpretations. This knowledge representation is also useful to carry out video analysis.

3 High Level Feature Extraction from Video Content

Knowledge representation is one of the fundamental aspects for implementing an inference system. The knowledge is acquired from both video content and users. The inference system is based on a fuzzy associative memory (FAM) model.

The proposed approach uses FAM to provide a framework which maps low level features through a group of fuzzy sets into either another group of fuzzy sets or a group of crisp sets. The output values are associated with high level features represented by concepts. These concepts are used for annotating video sequences. In addition, these concepts can be utilized for construing the semantic content of the video.

Knowledge representation denoted by $\Re$ establishes relations between elements from a space of low level feature values $\varGamma$ into a space of high level features $\varDelta$

$$\Re : \varGamma \mapsto \varDelta \tag{1}$$

3.1 Learning Stage

To overcome semantic gap this relations are initially acquired from one or more users acting as annotators in a *Learning stage*. In Fig. 1 this stage is illustrated.

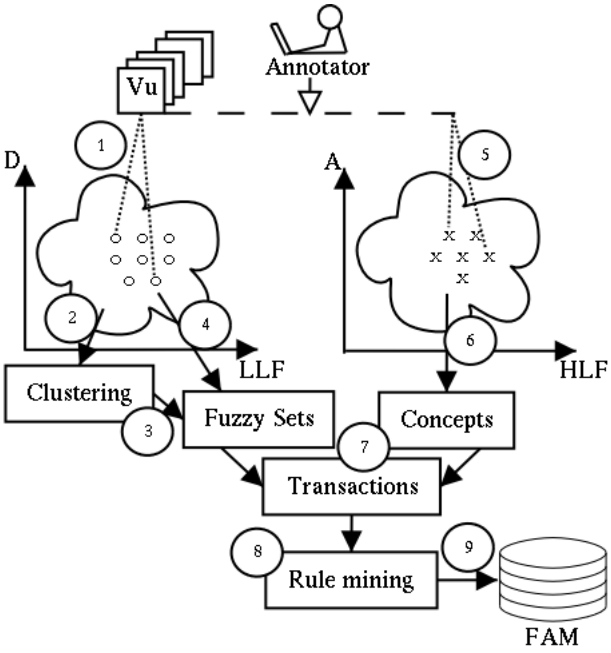


Fig. 1. Overview of the Learning stage. Each step is indicated by numbers and the dataflow by arrows. Vu, LLF, D, HLF, A and FAM stand for Video Unit, Low Level Feature, Descriptor, High Level Feature, Accuracy and Fuzzy Associative Memory, respectively

Each element $\gamma \in \mathit{\Gamma}$ represents low level features LLF extracted using one or more descriptors D from video units (Step 1). These video units can be still images, group of pictures or other sequences got from the streams.

Applying the clustering algorithm Fuzzy C-Means, the feature values are grouped into clusters to identify the boundaries of the fuzzy sets (Step 2). The fuzzy sets are automatically generated from these boundaries (Steps 3). These fuzzy sets constitute the inputs of the FAM (Step 4).

Each element $\delta \in \mathit{\Delta}$ corresponds to high level features HLF representing by concepts and provided by the annotators. These concepts are associated to the sample data and accuracy A is assigned (Step 5).

In this way, a group of either fuzzy sets or crisp sets, depending on the criteria of the annotators, is automatically generated. It constitutes the outputs of the FAM (Step 6).

The fuzzy sets represent the fuzziness in the mapping between a low level feature value and its interpretation. A fuzzy set contains elements that have varying degrees of membership in the set. The degree of membership is estimated using a function-theoretic form. This function maps elements of a fuzzy set $\widetilde{A}$ to a real value in the interval [0,1]. If a given element x in the universe, is a member of the fuzzy set $\widetilde{A}$, then this mapping is given by $\mu_{\widetilde{A}}(x) \in [0,1]$. As usual membership functions represent a possibility distribution.

Each fuzzy set represent an entry in a lexicon, which contains a set of user-specified words. Thus, it is possible to give meaning to each fuzzy set.

The set of rules for the FAM is generated applying a data mining technique on a set of transactions. Each transaction represents a combination of fuzzy sets and concepts as follows:

$$t_\gamma : (\mu(\gamma), \delta_\gamma) \tag{2}$$

where t_γ is a transaction for an element $\gamma \in \mathit{\Gamma}$, $\mu(\gamma)$ is a list of fuzzy sets associated to γ with memberships greater than zero and $delta_\gamma$ is the concept associated to the sample data where γ was extracted (Step 7).

The result of the data mining process is a set of frequent patterns representing association rules between elements of $\mathit{\Gamma}$ and elements of $\mathit{\Delta}$ (Step 8).

$$R : \mathit{\Gamma} \rightarrow \mathit{\Delta} \tag{3}$$

This result can contain non interesting rules for the inference system. Therefore, a filter is applied for selecting the most suitable rules (Step 9). The inference system contains rules of the form: “IF *condition* THEN *action*”. Each condition references low level features under the abstraction provided by the fuzzy sets. On the other hand, each action is linked to concepts representing high level features. Joining the inputs, outputs and rules, the FAM-based knowledge representation is ready.

3.2 Generalization Stage

Once a knowledge representation has been established, new video units can be automatically annotated. This is done by the generalization stage. In Fig. 2 this stage is illustrated.

Using one or more descriptors, low level features are extracted from video units (Step 1). These feature values are mapped into concepts applying the FAM-based knowledge representation. FAM uses an inference system which involves three basic components: fuzzification, fuzzy inference and defuzzification (Step 2).

The input of the fuzzification is a feature value and the output is a fuzzy value, which represents the degree of membership of the feature value to each fuzzy set. The conditions are instantiated according to the results of the fuzzification.

The fuzzy inference uses the knowledge rules base. Here, the conditions express instances of low level features and the actions link concepts. At this point, a number of rules can have different degrees of truth leading to competition

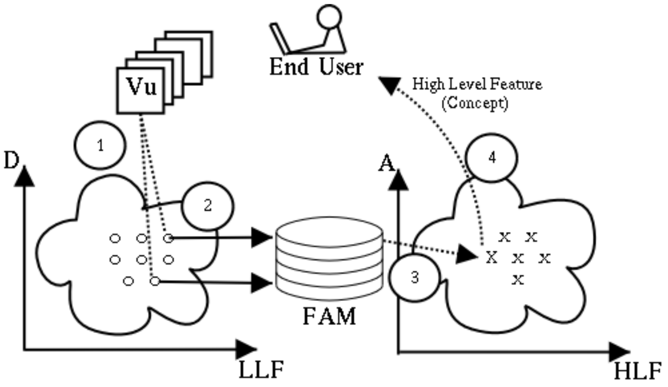


Fig. 2. Overview of the Generalization stage. Each step is indicated by numbers and the dataflow by arrows. Vu, LLF, D, HLF, A and FAM stand for Video Unit, Low Level Feature, Descriptor, High Level Feature, Accuracy and Fuzzy Associative Memory, respectively

between the results. Using an aggregation's operator the instances of the conditions are combined in order to determine the value of the rule. This value is used to determine the resulting actions. The procedure is repeated for all rules. Besides, it is possible that an output fuzzy variable has a fuzzy set as action in several rules. The composition's operator is used to determine the final value of this fuzzy set.

The defuzzification combines the fuzzy values of each output variable to obtain a real number for each one, which denotes the degree of possibility of the concept for annotating the corresponding video unit. In this module a weighted average method is used. It combines fuzzy values using weighted averages to obtain the resulting crisp value (Step 3).

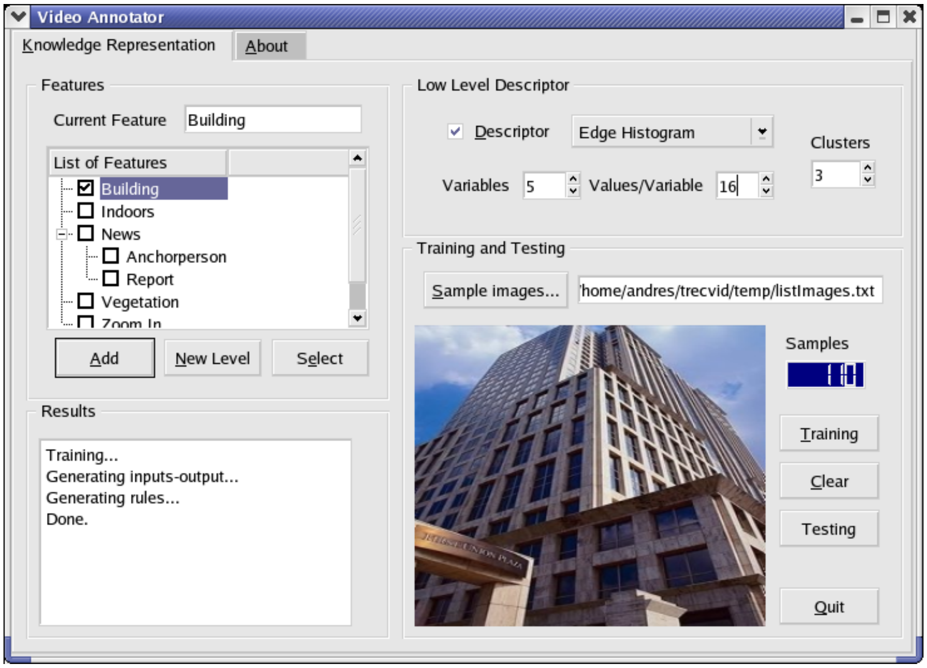
The concepts referenced by each output variable are selected depending on the instance of the variable. The result consists of a list of concepts with degrees of possibility. This list is delivered to the user (Step 4).

4 Experimental Results

The proposed system was implemented in C++ on a Linux platform. Fig. 3 shows a screenshot of the graphic user interface for the system. A database of over 100 randomly chosen MPEG2 video clips with durations between 2 and 6 minutes was used to evaluate the proposed approach.

Several experiments were run to test the knowledge representation approach. Only selected results for news clips showing video sequences containing *anchorperson* or *report* are reported in this section. These two concepts were linked with the temporal descriptors *shot length distribution* and *shot activity*, which were extracted from the sample videos in the compressed domain. Table 1 and

Fig. 4 show the settings used in the knowledge representation of these temporal descriptors.



Video Annotator

Knowledge Representation | About

Features

Current Feature: Building

List of Features
- ☑ Building
- ☐ Indoors
- ☐ News
  - ☐ Anchorperson
  - ☐ Report
- ☐ Vegetation
- ☐ Zoom In

Low Level Descriptor

☑ Descriptor: Edge Histogram

Variables: 5 Values/Variable: 16

Clusters: 3

Training and Testing

Sample images... 'home/andres/trecvid/temp/listImages.txt

Results

```
Training...
Generating inputs-output...
Generating rules...
Done.
```

Fig. 3. Screenshot of a prototype implementing the approach. The high level features are organized into a tree-based structured. MPEG-7 low level features are extracted from the frames. Each descriptor has a set of parameter settings. The number of cluster determines the amount of fuzzy sets automatically generated. Still images are taken as samples for training the system and generating the knowledge representation as a part of the Learning stage. A testing option is provided in order to evaluate the Generalization stage

Table 2 depicts some rules selected from the results of the association rule mining process. The support of these rules is low due to the amount of transactions generated from the sample data. The minimum confidence (*minconf*) was fixed in 1 or maximum confidence. The minimum support (*minsup*) was varied from 1 towards 0 using an interval of 0.01.

Two stop conditions were defined to optimize the computing time. The first is when the *minsup* reach its lower limit before zero. The second is given by a *neck condition*, which is defined as the point where the number of generated association rules is too big in comparison with the amount of new patterns that can be used in the inference process. In Fig. 5 an example of the *neck condition* is shown.

Table 1. Knowledge representation boundaries for the temporal descriptors. The first column indicates the descriptor used to define the Universe of the fuzzy sets. The second column lists the names of the fuzzy sets referencing to words in the lexicon. The remaining columns indicate the boundaries of the fuzzy sets

Universe	Fuzzy Set	b_1	b_2	b_3
Shot Length Distribution	$\widetilde{A}$=short	0.2	0.35	0.5
	$\widetilde{B}$=mid	0.2	0.35	0.5
	$\widetilde{C}$=long	0.1	0.25	0.4
Shot Activity	$\widetilde{A}$=low	0.3	0.50	0.7
	$\widetilde{B}$=mid	0.1	0.25	0.4
	$\widetilde{C}$=high	0.1	0.25	0.4

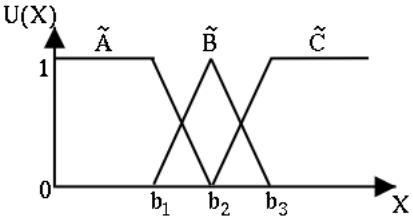


Fig. 4. Fuzzy sets for the temporal descriptors. b_1, b_2 and, b_3 are the boundaries of the fuzzy sets $\widetilde{A}$, $\widetilde{B}$ and $\widetilde{C}$. X corresponds to a feature and $\mu(X) \in [0, 1]$ its membership function

Table 2. Sample of inference rules obtained applying association rule mining. Only rules with *minconf* = 1.0 were considered. *minsup*$\in (0, 1]$ with $\varDelta = 0.01$

Support	Condition			Action
0.10	sld_shorts is low	AND	sld_longs is high	Labelled as anchorperson
	sld_longs is high	AND	sa_low is high	Labelled as anchorperson
	sld_longs is high	AND	sa_mid is low	Labelled as anchorperson
0.08	sld_longs is low	AND	sa_mid is high	Labelled as report
	sld_midis is high	AND	sa_low is low	Labelled as report
	sld_shorts is mid	AND	sa_mid is high	Labelled as report

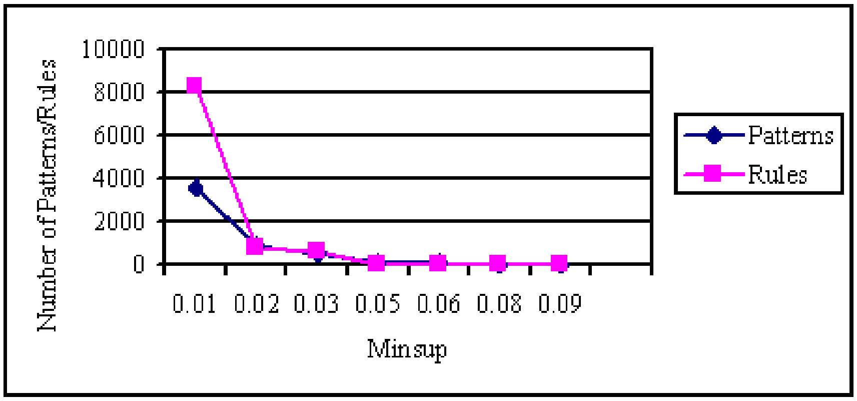


Fig. 5. Neck condition. Shifting *minsup* from 1 to 0, a neck condition is detected near to 0.02

Some representative results for the Generalization stage are summarized in the following three tables. The first column shows the names of fuzzy sets for each temporal descriptor. The second column corresponds to feature values extracted from the key-frames of the video sequence. The next three columns depicts the degree of truth of each value to the corresponding fuzzy sets estimated using the membership functions. Table 3 shows a case where the data match exactly the pattern of the concept *anchorperson*.

Table 3. Annotation results for a randomly selected news clip *News032*

	News032	$\widetilde{A}$	$\widetilde{B}$	$\widetilde{C}$
sld_shorts	0.000	1.000	0.000	0.000
sld_midis	0.382	0.000	0.788	0.212
sld_longs	0.618	0.000	0.000	1.000
sa_low	0.655	0.000	0.225	0.775
sa_mid	0.153	0.645	0.355	0.000
sa_high	0.191	0.390	0.610	0.000
Ground truth	anchorperson			
Fuzzy inference	anchorperson =		1.000	
	report =		0.000	

Table 4 presents a case where the data matched exactly the pattern of the concept *anchorperson* and partially the pattern *report*. It means the inference system is detecting a low possibility for the occurrence of this concept in the characteristic of the values.

Table 4. Annotation results for a randomly selected news clip *News081*

	News081	$\widetilde{A}$	$\widetilde{B}$	$\widetilde{C}$
sld_shorts	0.200	1.000	0.000	0.000
sld_midis	0.143	1.000	0.000	0.000
sld_longs	0.657	0.000	0.000	1.000
sa_low	0.418	0.410	0.590	0.000
sa_mid	0.292	0.000	0.718	0.282
sa_high	0.289	0.000	0.738	0.262
Ground truth	anchorperson			
Fuzzy inference	anchorperson =		1.000	
	report =		0.212	

Table 5 displays a case where the data matched partially the pattern of the concept *report*. It means the inference system is detecting a high but not total possibility for the occurrence of this concept in the characteristic of the values. The system did not detect possibility for the concept *anchorperson*.

Table 5. Annotation results for a randomly selected news clip *News138*

	News138	$\widetilde{A}$	$\widetilde{B}$	$\widetilde{C}$
sld_shorts	0.333	0.111	0.889	0.000
sld_midis	0.544	0.000	0.000	1.000
sld_longs	0.122	0.852	0.148	0.000
sa_low	0.317	0.916	0.085	0.000
sa_mid	0.380	0.000	0.131	0.869
sa_high	0.302	0.000	0.652	0.348
Ground truth	report			
Fuzzy inference	anchorperson =		0.000	
	report =		0.916	

The accuracy of the approach for these results was evaluated and it is summarised in Table 6.

5 Conclusions and further work

In this paper, video analysis refers to semantic interpretations of the video content. These interpretations associated to low level features must be structured in an effective and meaningful manner. The presented approach uses a tree-based data model. This is a first step towards a graph-based model which is suitable for processing data in video systems.

Table 6. Recall and Precision results of the annotation process

Keywords	Detects	Missed Detections	False Alarms	Recall	Precision
Anchorperson	46.65	1.36	2.48	0.97	0.95
Report	25.74	6.28	1.32	0.80	0.95

Besides, graph-based structures are appropriated to represent relationships among video annotation objects. These objects are temporally and spatially linked to video segments and contain descriptors characterizing video content.

The knowledge representation based on a fuzzy associative memory model has several advantages: it is modular and consequently easy to adapt in software systems, its hardware implementation is feasible, and it is relatively simple and consistent with the way human reasoning works.

The interaction with users is required because of the complexity of the semantic-based content annotation process, which involves data, metadata and user's interpretations.

The concepts used for annotating video sequences can be utilized for construing semantic content in videos. Due to the fact that concepts are described by keyword and key-phrase only, this approach is restricted to weak semantic inferences.

Acknowledgments

The research leading to this paper was done within the framework of the Network of Excellence SCHEMA.

References

1. Niblack, W., Barber, R., Equitz, W., Flickner, M., Glasman, E., Pektovic, D., Yanker, P., Faloutsos C., Taubin, G.: The QBIC Project: Querying Images by Content Using Color, Texture, and Shape. In: Proc. SPIE. Vol. 1908. (1993) 173-187
2. Smith, J.R., Chang, S.-F.: VisualSeek: A Fully Automated Content-Based Image Query System. In: Proc. of the ACM Int'l Conference on Multimedia. (1996)
3. Ponceleon, D., Srinivasan, S., Amir, A., Petkovic, D., Diklic, D.: Key to Effective Video Retrieval: Effective Cataloguing and Browsing. In: ACM Multimedia. (1998) 99-107
4. Zhong, D., Chang, S.-F.: An Integrated Approach for Content-Based Video Object Segmentation and Retrieval. In: IEEE Trans. on Circuits and Systems for Video Technology. Vol. 9. No. 8. (1999) 1259-1268
5. Ohm, J.-R., Bunjamin, F., Liebsch, W., Makai, B., Müller, K., Smolic, A., Zier, D.: A Multi-Feature Description Scheme for Image and Video Database Retrieval. In: Proc. IEEE Multimedia Signal Processing Workshop. Copenhagen (1999) 123-128

6. Sawhney, H.S.: Motion Video Annotation and Analysis: An Overview. In: Conference Record of The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers. Vol. 1. (1993) 85 -89
7. Bobick, A.F.: Representational Frames in Video Annotation. In: Conference Record of The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers. Vol. 1. (1993) 111-115
8. Benitez, A.B., Smith, J.R.: New Frontiers for Intelligent Content-Based Retrieval. In: Proc. SPIE. Vol. 4315. San Jose, CA (2001)
9. Salway, A., Ahmad, K.: Multimedia Systems and Semiotics: Collateral Texts for Video Annotation. In: IEE Colloquium on Multimedia Databases and MPEG-7 (Ref. No. 1999/056). (1999) 1-7
10. Radev, I., Pissinou, N., Makki, K.: Film Video Modeling. In: Proc. Workshop on Knowledge and Data Engineering Exchange (KDEX '99). (1999) 122 -128
11. Hibino, S. Rundensteiner, E.A.: A Visual Query Language for Identifying Temporal Trends in Video data. In: Proc. Int'l Workshop on Multi-Media Database Management Systems. (1995) 74 -81
12. Dorado, A. and Izquierdo, E.: An Approach for Supervised Semantic Annotation. In: Izquierdo, E.(ed.): Digital Media Processing for Interactive Services. Proc. 4th European Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS03). World Scientific, London (2003) 117-121
13. Jang, S.-W., Moon, C.-H., Choi, H.-I.: Shot Transition Detection with FAM-Based Fuzzy Inference. In: Proc. IEEE Int'l Conference on Fuzzy Systems (FUZZ-IEEE '99). Vol. 2. (1999) 869 -874
14. Dimitrova, N.: The Myth of Semantic Video Retrieval. In: ACM Computing Surveys. Vol. 27. No. 4. (1995)
15. Stefik, M.: Introduction to Knowledge Systems. Morgan Kaufmann Publishers Inc. (1995)
16. Intille, S.S., Bobick, A.F.: Closed-World Tracking. In: Proc. IEEE Fifth Int'l Conference on Computer Vision. (1995) 672-678
17. Naphade, M.R., Huang, T.S.: A Probabilistic Framework for Semantic Indexing and Retrieval in Video. In: IEEE Int'l Conference on Multimedia and Expo (ICME 2000). Vol. 1. (2000) 475 -478
18. Davis, M.: Media Streams: An Iconic Visual Language for Video Annotation. In: Proc. IEEE Symposium on Visual Languages. (1993) 196 -202
19. Ki-Wook, K., Ki-Byoung, K., Hyoung-Joo, K.: VIRON: An Annotation-Based Video Information Retrieval System. In: Proc. 20th Int'l Conference on Computer Software and Applications (COMPSAC '96). (1996) 298-303

A Novel Method to Describe Multimedia Documents

Yan Jianfeng, Zhang Yang, Li Zhanhuai

Department of Computer Science & Engineering,
Northwestern Polytechnical University,
710072 Xi'an, Shaanxi, P. R. China
jfyan@mail.nwpu.edu.cn,zhangy@co-think.com,lizhh@nwpu.edu.cn
http://www.nwpu.edu.cn

Abstract. The most important thing of multimedia information system is to provide overall and detailed description of the content of multimedia data. Currently, the descriptions of multimedia documents, such as MPEG-7, have some shortage in providing the description of this information in different layers and different granularities. In this paper, we locate the content information of the multimedia documents into three proper layers and define MDU (Multimedia Document basic Unit) to collect this information in different granularities. And then, we discuss the relations among MDUs. Combining the definitions above, we can formalize the presentation of the multimedia documents and multimedia document database. An example is provided to explain how to use this presentation method, and we also compare this method with related works. Lastly, we also explain how to operate this model.

1 Introduction

In recent years, with the rapid development of computer technology, especially powerful workstations, high capacity digital storage systems (e.g. optical discs) and the rapid reduction in the cost of these devices, it is possible to store a great amount of multimedia data in computer system. At the same time, along with the widely used Internet, the research of multimedia data management becomes more and more important [1]. The Information Retrieval (IR) system that manages the alphanumeric data is changing to the Multimedia Information Retrieval (MIR) system whose purpose is to storage and retrieval multimedia data. Many authors assert that MIR system is fundamentally different from traditional IR system, and these traditional methods and techniques are therefore not well suited to MIR system [2,3].

The kernel of the MIR system is to find the information user interested from a great deal of multimedia data [4]. To achieve this purpose, the database of the MIR system must present the content of the multimedia document completely and strictly, and this kind of presentation does not only contain the content based on the structured features (e.g. spatial relations) but also on the semantic features (e.g. the information “shake hands”)[5,6].

2 Related Works

Multimedia data is more complicated than structural data. Firstly, the data structure of multimedia data is semi-structure (e.g. HTML, XML) or no-structure (e.g. texts, images, audios and videos)[4]. The information managed by MIR system must be carried by structural data, so the biggest challenge for presenting the content of multimedia data is to make them structural. In recent researches, lots of methods, which use many kinds of data types, have been proposed to solve this problem [7,8], but the multiplex data types make it difficult for MIR system to communicate with each other. Secondly, multimedia data always carry a great amount of content information. For example, the presentation of a image data could contain the file type, texture, color and shapes of the objects in the image. Lastly, the process to cognize the semantic information of the multimedia data depends on the knowledge from different backgrounds. Even for the same multimedia data, different viewers pay attention to different semantic contents [9].

To uniform the multiplex data types, the Moving Picture Experts Group established the MPEG-7 [10], which provides a standard method to present the multimedia document using XML documents. Currently, the biggest shortcoming for MPEG-7 is that it cannot present the multimedia document in different layers and in different granularities.

Though many research projects have been done in research field of multimedia data management [11,12], there are still many progresses need to be made [9]. One hand, we need a uniform model to describe all kinds of media data. Ozkarahan [13] found that the multimedia documents is different of traditional documents in complex relations, multiple objects, user interface and presentation. Along with the Web multimedia information more and more, more and more documents change from single image, text data to mixed media documents. It means traditional methods and techniques used in single media databases are therefore not well suited. On the other hand, many of present models cannot provide overall and detailed description of the content of multimedia data, so the query is either query-by-example or query-by-subject[11]. It is important to build a model to contain all information, including structural and semantic ones. Certainly, there are several models [14,15], which provide the management of multimedia data. Comparing with them, our works pay more attention to the description of the overall content information.

Usually, the content information of the multimedia data could be located into three layers and there are tight mapping relations among them:

1. Physical layer: presenting the information of raw-data and meta-data. Raw-data is the binary data stream of the multimedia data and meta-data contains the generation information, formatting information and usage information of the multimedia data.
2. Structural layer: presenting the structural objects (e.g. a rectangle) and the structural relations among objects, such as the spatial, temporal and spatiotemporal relations among objects [7].
3. Semantic layer: presenting the semantic objects (e.g. the sun) and the semantic relations among objects [8].

We will describe a novel model to present the content of the multimedia document in all different layers and different granularities. The most distinctive contribution of our model is to provide a method to describe the integral information of multimedia document. The “integral” means not only the raw-data and mete-data but also structural and semantic can be represented. And, we also explain the query method to operate all data descriptions above. So, theoretically speaking, our model can be used by both query-by-example and query-by-subject.

In section 3, we introduce our data model and build the multimedia documents database based on it normally; in section 4, an example is used to explain how multimedia documents are described; in section 5, an important function and several operations are introduced to explain how to query the multimedia document database; in last section, we conclude what we have done and present what will be done in the future.

3 Modeling the Multimedia Document Database

3.1 Multimedia presentation basic units and the mapping relations

Generally speaking, there are two kinds of multimedia documents, the simple multimedia documents and the complex multimedia documents. The former comprise an instance of one type of media while the latter comprise several instances of one or several types of media. The MPEG-7 presents the multimedia documents by the object-oriented idea [16,17]. For example, an image is presented by a set of objects and a set of relations among them and every object has several properties including media features, visual features and semantic features. The relations are described by the objects’ hierarchy figures and the ER figures.

From the introduction above, we can understand that the key idea of MPEG-7 is to format the multimedia data into objects and relations among these objects, so we can collect the objects whose have some relations among them into one set and add several proper presentation as a Multimedia Document basic Unit (MDU). MDU is defined as following:

Definition 1. A **Multimedia Document basic Unit (MDU)** is a 3-tuple *m: <MO,RC,R>*, where *MO* is a set of objects described by the MDU; *RC* is a set of all kinds of relations. *R* is a subset of $RC \times 2^{MO} \times 2^{MO}$, and it means the relations among the objects.

For notation convenience, we give these assumptions: 1.If there are no semantic confusion we do not distinguish *{X}* from *X* (*X* is a object or a tuple), for instance, if there is only one object *O* in the *MO* of a MDU, we write *<O, Ø, Ø>* for the MDU of this object. 2.We overload the operator “.”. It cannot only get a heft from a tuple but also a property or a function from an object.

Each MDU can describe a *block* of information in one layer. The *block* here means that if there are several relations between objects within two different MDUs, these two MDUs must be combined together. Each object in *MO* contains several properties, functions and constraints. The properties describe the features of an object. The values of the properties in a certain layer represent the values of the features in the same layer.

The functions are used to achieve the values and the different arithmetic may get the different values of one same property. The constraints are used to design the important limitations.

3.2 Intra-MDU relation

The intra-MDU relation is based on *RC*. Some certain relations among objects have some special properties. For instance, the temporal relation ***Parallel*** is an equivalence relation; the relation ***Similar*** is a fuzzy equivalence relation, and so on. Those relations imply some potential information. For example, *Parallel*(O_1,O_2) and *Parallel*(O_2,O_3) imply *Parallel*(O_2,O_1), *Parallel*(O_3,O_2), *Parallel*(O_1,O_3) and *Parallel*(O_3,O_1). It is essential to divide the intra-MDU relations into several kinds. It will enhance the performance of the MIR system and reduce the consumed cost of the resources. The operations upon *RC* need to be constructed on formalization of the relations.

Definition 2. The **intra-MDU relation** of a MDU *m* for certain kind of relations is described by a directed net (r,V,A,W), where:

r is the label of the net, which means one certain relation. $r\in m.RC$.

V is the set of nodes, $\{O_1,O_2,O_3 ,\ldots, O_n\}$ Í $m.MO$.

A is the set of the edges. For each $r(O_i,O_j) \in m.R(O_i,O_j\in V)$, there is an edge.

W is the set of weights. Each weight represents *immediate degree* of one edge in A.

Commonly, we define the *immediate degree* as real number 1 to present there is the relation r between O_i and O_j, while If r is fuzzy relation we define the *immediate degree* as a real number between 0 and 1. If there are two relation r and r' which are opposite, we can find one' directed net from another's because they are same in the shape and opposite in the edges.

From the definition above, if there are two objects O_a and O_b in the set of nodes of a certain directed net N labeled by relation r, and there is a subset N' whose edge set A' is $\{(O_a, O_i),(O_i, O_{i+1}), (O_{i+1}, O_{i+2}),\ldots,(O_{i+j}, O_b)\}$ exists in N, In another word, if O_a to O_b can be arrived, there is a intra-MDU relation r between O_a to O_b. The corresponding ***deduced degree*** between them is defined as β^{ab}.

According to the definition, the *immediate degrees* among objects are certain kinds of ***deduced degrees*** whose objects have an edge between each other. Between no directly adjacent objects, the ***deduced degree*** needs be spliced by some other *immediate degrees*. Splicing operator is defined as "Θ". And $\beta^{xz}=\beta^{xy}\Theta\beta^{yz}$ express the splicing of random two kinds of ***degrees***. Analyzing the feature of relations among objects, this operator must have following characteristics:

a). $(\beta^{xy}\Theta\beta^{yz})\ \Theta\beta^{zk}=\beta^{xy}\Theta(\beta^{yz}\Theta\beta^{zk})$

b). $0\leq\beta^{xy}\Theta\beta^{yz}\leq 1$

c). There are β^{xy} and β^{yz} , and both are not 0 or 1, then $\beta^{xz} < max(\beta^{xy},\beta^{yz})$

d). There is β^{xy}=1, then $\beta^{xz}=\beta^{yz}$, and There is β^{yz}=1, then $\beta^{xz}=\beta^{xy}$

e). There is β^{xy}=0, then β^{xz}=0, and There is β^{yz}=0, then β^{xz}=0

The characteristic a) is the algebra quality for "Θ", means the ***degree*** is incorporable and steady. The rests are boundary quality: The characteristic b) points out that the ***degree*** is regular. The characteristic c) elucidates that if the ***degree*** has been transmitted, the relations between objects become weak; The characteristic d) elucidates that if two objects have one relation completely, the ***degree*** of the former with the third object is decided by that of the latter with the third, The characteristic e) means if two objects have not one relation, the former will be irrelevant to the objects which can be reached by the latter.

A lot of different operations of the Splicing operator can be constructed according with the above requirement, and those should be defined according to the effective demands of the MIR system. we define a kind of simple linear operation as follows:

$$\beta^{xy}\Theta\beta^{yz}=\beta^{xy}\times\beta^{yz} \qquad (\mathbf{1})$$

Obviously, this definition satisfies the requests of above five characteristics.

3.3 Inter-MDU relation

As discussed above, we know that using MDUs we can collect the objects and several relations among these objects as a ***block*** of information. Using splicing operator "Θ", we can extend the relations and measure them with ***immediate degrees*** or ***deduced degrees***. Above all, the content information of multimedia documents can be described in the respective layers. There is still a problem, how to present the connection of MDUs between different layers. In MIR system, we have to answer some inter-layered query, such as “Find the binary data streams of images including a event that the sun is rising from sea”. Firstly, we should find all those images containing the sun and sea, and this information is described in semantic layer. Secondly, we must refine those images to find the up-down spatial relation between the sun and sea, and this information is described in structural layer. Then, we can get the binary data streams user needed in physical layer. So, it is necessary to describe the relations among MDUs between layers.

Definition 3. The **inter-MDU relation** between two MDUs m_1 and m_2 is a set of mappings $\lambda: O\rightarrow O'$, marked by $n(m_1, m_2)$. O and O' are the set of objects which satisfy O Í $m_1.MO$ and O'Í $m_2.MO$.

MDU provides the data structure for presenting multimedia data. Considering the different abstraction layers, the description of a multimedia document should contain many MDUs, which are located in all of the three layers, and the relations among all theses MDUs.

3.4 Formalization of the multimedia document database

Combining the definitions above, we can formalize the presentation of the multimedia document.

Definition 4. A **Multimedia Document (MD)** is a 4-tuple $M^D: < M_p, M_c, M_s , N >$. M_p, M_c and M_s is the set of MDUs. They denote the set of MDUs of physical layer, structural layer and semantic layer separately. N is the set of $n(m_i, m_j)$ $(m_i\in M_p\cup M_c\cup M_s,\ m_j\in M_p\cup M_c\cup M_s)$, describing the mapping relations among the MDUs.

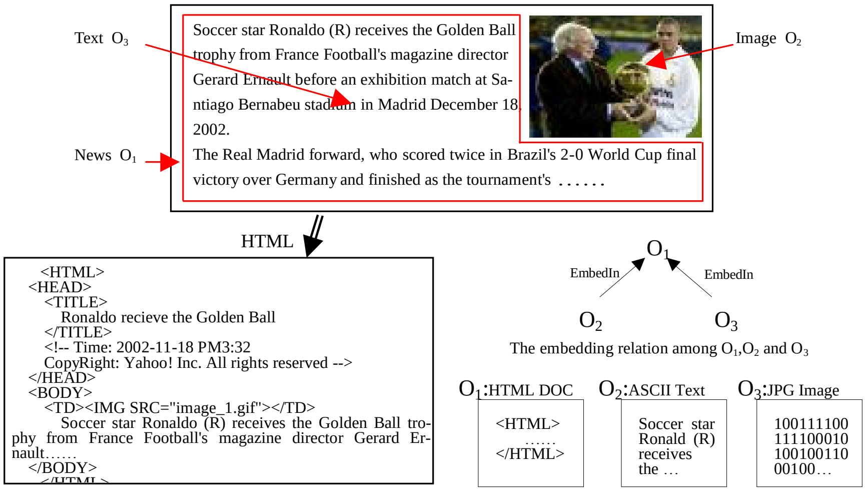
Generally speaking, because different application have different requirement, even in a single abstraction layer there are several MDUs for describing a multimedia document. For example, in the physical layer of a text, two MDUs are used to present two kinds of coding method for this text. In the structural layer of an image document, one MDU maybe used to describe the color information of this image, and another MDU maybe used to describe the shape information of objects in this image. If we need to describe a complex multimedia document, which contains *n* simple multimedia document, then at least we need *n* MDUs in the structural layer.

From the above definition, we can formalize the multimedia document as a structured data type MD. Based on this, the following definition will build a multimedia database.

Definition 5. A **Multimedia Document Database (MDD)** is a set of MDs, M: $\{M^D{}_1, M^D{}_2, M^D{}_3, \ldots, M^D{}_n\}$, where every $M^D{}_i$ means one of the multimedia documents which should be managed by this MDD.

4 Layered Description of Multimedia Documents

4.1 Description in physical layer



Soccer star Ronaldo (R) receives the Golden Ball trophy from France Football's magazine director Gerard Ernault before an exhibition match at Santiago Bernabeu stadium in Madrid December 18 2002.

The Real Madrid forward, who scored twice in Brazil's 2-0 World Cup final victory over Germany and finished as the tournament's ……

```
    <HTML>
  <HEAD>
     <TITLE>
        Ronaldo recieve the Golden Ball
     </TITLE>
     <!-- Time: 2002-11-18 PM3:32
     CopyRight: Yahoo! Inc. All rights reserved -->
  </HEAD>
  <BODY>
     <TD><IMG SRC="image_1.gif"></TD>
        Soccer star Ronaldo (R) receives the Golden Ball tro-
phy from France Football's magazine director Gerard Er-
nault……
  </BODY>
    </HTML>
```

Fig. 1. An example of a complex multimedia document including the media instances and several relations among them in physical layer

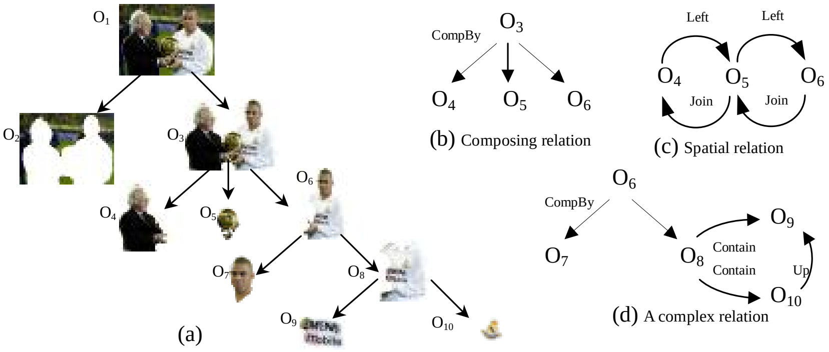
In physical layer, $m_i.MO(m_i$ Í $M_p)$ is a set of media instances $\{O_1, O_2, O_3 \ldots\ldots\}$ in complex multimedia document. Especially, the MDU m_i is a 3-tuple $<O, Ø, Ø>$ if we present a simple document. The example shown in figure 1(a) is a complex document composed by a paragraph of text and an image (about the same news). As far as the attention in physical layer, the document O_1 is composed by an image O_2 and a paragraph of text O_3 with the special spatial relation and temporal relation between them. The graph figue 1(b) shows these relations.

Without other information included, a MDU named m_p is the description of the multimedia document shown above in physical layer.

m_p:<{O_1,O_2,O_3},
{ *EmbedIn* },
{< *EmbedIn*,O_2,O_1>,< *EmbedIn*,O_3,O_1>}>

One of the important properties of the objects in physical layer is the raw-data stream and accordingly the function is defined to code and decode this stream. Another properties is used to present the meta-data, including the generation information (e.g. the generator, generating time and place), formatting information and usage information (e.g. copyright). The constraints of objects in physical layer mean the transition velocity, the synchronization relations, and so on.

4.2 Description in structural layer



(a)

(b) Composing relation

(c) Spatial relation

(d) A complex relation

Fig. 2. The objects of the image in structural layer and several relations among them

The set M_c in structural layer describe the structural objects and the structural relations among objects and One of the media instances should contain one MDU at least. What showing in figure 2 is the structural information of the image shown in figure 1(a), which should be described after the region-dividing and the target-extracting processing. Figure 2(a) shows the spatial structural relations, and the set of objects {O_1,O_2,…O_{10}} is used to present ten objects which must be described in MDU. Comparing with the object O_4, the object O_6 is divided into several detailed objects. It presents that the description can have different granularities determined by the different applications and realized by the mapping between the structural layer and semantic layer. The figure 2(b) present the composing relation among objects. The figure 2(c) shows the spatial relation. The relation ***Left*** is a partial relation, and the information *Left*(O_4,O_6) implied in the figure. The ***Join*** is a symmetrical relation, and there imply the information $Join(O_4,O_5)$ and $Join(O_5,O_6)$. The figure 2(d) depicts an other complex relation which includes several kinds of relations. The ***Contain*** means that the object O_8 contains O_9 and O_{10}.

According to the figure 2, there is a MDU to describe the objects of the image and several relations among them in structural layer. There may be another MDU $m_c{}'$ to describe the information of text "Soccer star Ronaldo (R) receives the Golden Ball ……" in structural layer

$mc:<\{O_1,O_2,O_3,O_4,O_5,O_6,O_7,O_8,O_9,O_{10}\},$
$\{CompBy, Left, Join, Contain, Up\},$
$\{<CompBy,O_3,\{O_4,O_5,O_6\}>,<CompBy,O_6,\{O_7,O_8\}>,<Left,O_4,O_5>,<Left,O_5,O_6>, <Join,O_6,O_5>,<Join,O_5,O_4>,< Contain,O_8, \{O_9,O_{10}\}>,<Up,O_{10},O_9>\}>$

In structural layer, the properties of the objects depend on the raw data stream in physical layer and the values of them correspond to the media feature' values and the statistical values. The functions are the great deal of the text matching arithmetic, the image processing arithmetic and the video dividing arithmetic that can obtain those values. The constraints are used to design the important limitations, for example, one especial property must be calculated before another one.

4.3 Description in semantic layer

The description of the semantic layer's information is more complex. There are many reasons. Firstly, it needs complex data types to describe the semantic information; more above, some information especially the abstract conceptions, just like "happiness", cannot be described properly at all. Secondly, there are rich abstract levels in semantic information and it is hard to describe all information in these levels. Lastly, the semantic information usually depends on the special backgrounds, then there may be different explanations to the one multimedia document. Only the experts can resolve these ambiguities.

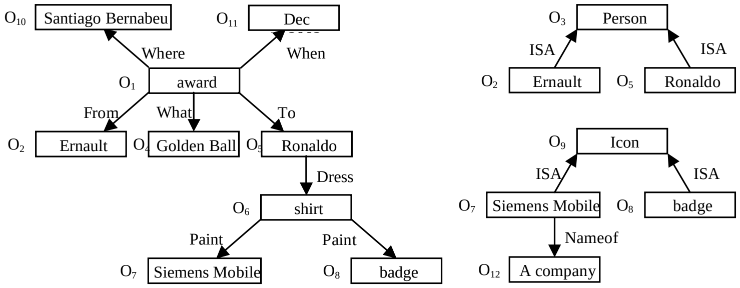


Fig. 3. The objects of the image in semantic layer and several relations among them

To describe the semantic information, the MPEG-7 provides many **Descriptors** (Ds) and **Description Schemas** (DSs) [18]. They can be used as the unified data structures, but there are several problems. For example, the semantic information of O_3 in figure 2 may be "Ernault is awarding Golden Ball to Ronaldo", or "Ernault is awarding something to Ronaldo", or "Someone is awarding something to Ronaldo", and so on.

If we only use the Ds and DSs to enumerate those great deal of information without distinguish the abstract levels. It may be “Data Blast”. So we use two kinds of constraints to limit the description, one named media-structural constraint and the other abstract-level constraint. The former constrains the structural information, For example, there is a media-structural constraint in the event object O_1 “awarding” as “*X* be awarding *Y*”(*X* and *Y* are variables); another more strictly media-structural constraint maybe “in *P* place, *Q* time, *X* be awarding *Y* to *Z*”. The latter pays attention to the different abstract levels. For example, there is an abstract-level constraint in the person object O_3 “Ernault” as “Ernault has a generalization *X*”(*X* is a variable).

From the figure 3 above, we can get a new MDU.

$ms{:}<\{O_1,O_2,O_3,O_4,O_5,O_6,O_7,O_8,O_9,O_{10},O_{11}\},$

$\{Where, When, From, What, To, Dress, Paint, ISA, Nameof\},$

$\{<Where,O_1,O_{10}>,<When,O_1,O_{11}>,<From,O_1,O_2>,<What,O_1,O_4>,<To,O_1,O_5>,<Dress,O_5,O_6>,<Paint,O_6,O_7>,<Paint,O_6,O_8>,<ISA,O_2,O_3>,<ISA,O_5,O_3>,<ISA,O_7,O_9>,<ISA,O_8,O_9>,<Nameof,O_7,O_{12}>\}>$

In semantic layer, each object in MDU is a basic semantic entity and a relation among objects describes an integral semantic unit. The properties, functions and constraints in semantic layer should be different along with the different background knowledge.

4.4 Mapping relations of the multimedia documents

There are many mapping relations among MDUs. The mapping from physical layer to structural layer will realize the feature extracting from raw-data, which correspond to the definition of the SegmentDS and RegionDS in MPEG-7. The mapping from structural layer to semantic layer will realize the contact between the information background independent and the information background dependent but the feature extracting, for example, the mapping of a circle to the sun. Other important mappings consist in the same layers. This kind of mapping describes the aggregation semantic and media feature links. The former presents the aggregation semantic among the objects in different MDU. Using this one, we can aggregate several objects into a new object and use another MDU to describe the information needed. The latter present the links between objects. For example, there is such link in a sunset image and an article having the word “sun”. Some correlative discussion can be read in [19].

Adding the mapping relations with MDU, an overall representation of the example multimedia document is $M^D_e{:}<m_p, \{m_c ,m_c'\}, m_s , \{n(m_p ,m_c),n(m_p ,m_c'),n(m_c ,m_s)\} >$.

5 Query the Multimedia Document Database

In this section, we explain how to address the multimedia document database M: $\{M^D_1, M^D_2, M^D_3,\ldots, M^D_n\}$, where every M^D_i present the content information of one of the multimedia documents in different layers and different granularities fully. Before the query operation defined, we have to express one important function.

Contain function

There are two MDUs x and y, if and only if $x.MO$ Í $y.MO$, $x.RC$ Í $y.RC$ and $x.R$ Í ′$y.R$(the Í ′ operator means a *fuzzy_belong_to* operation which use the intra-MDU relations to complete the information of MDUs, detailed discussion can be seen in [19]), function *Contain(x,y)* return true, otherwise, *Contain(x,y)* return false. Actually, *Contain* function present that whether the information of x is contained by y. We can overload the *contain* function to exam the inclusive relation between a MDU and a MD. That is, if $Contain(x,M^D)$(M^D is a MD) is true, there must exist at least one MDU $y(y\in M^D.M_p \cup M^D.M_c \cup M^D.M_s)$ which make *Contain(x,y)* return true. Obviously, it means that M^D contains the information of x completely.

Example 1. Test the contain function of *<{Golden Ball}, Ø, Ø >* and m_s.

Obviously, because O_4 *=Golden Ball* and Ø could belong to every set, we know *Contain(<{Golden Ball}, Ø, Ø >,* m_s*)*. It means that m_s contains the information *Golden Ball*.

We propose that the input is a MDU n, then the query will be processed as below:

5.1 Select the relational multimedia documents

The *select operation* selects multimedia documents from a multimedia document database that contain the information of a given MDU. We use the lowercase Greek letter sigma "σ" to denote this operation.

Definition 6. The **select operation** $\boldsymbol{\sigma}$ find a set of MDs from a MDD, denoted as $N = M_{\sigma}(n)$, where: n is the given MDU, and for each $M^D(M^D\in M\ )$, if function $Contain(n,M^D)$ returns true, $M^D\in N$.

5.2 Shrink to tidy multimedia documents

The *shrink operation* shrinks every multimedia document from N above to tiny multimedia document. We use the lowercase Greek letter sigma "ς" to denote this operation.

Definition 7. The **shrink operation** ς allows us discard all MDUs except those MDUs, which contain n, from every multimedia document in N , denoted as $H = N_{ς}(n)$, where: for each $M^D(M^D\in N\ )$, firstly, we collect every MDU m, which makes $Contain(n,m)$ return true, to a new MD as $M^{D\prime}$. Then, we also add every MDU m' from M^D, which satisfy $n(m, m')\in M^D.N$ or $n(m', m)\in M^D.N$, to $M^{D\prime}$. Lastly, we get a new MD as $M^{D\prime\prime}$, $M^{D\prime\prime}\in H$.

5.3 Project to a certain state

The *project operation* projects every tiny multimedia document from H above to certain layer according to users requirement. Actually, what we need to do in this step is only to discard all MDUs except those in that certain layer, Of course, correlative mappings between MDUs will be discarded too. We use the lowercase Greek letter sigma "π" to denote this operation.

Definition 8. The **project operation π** allows us discard all MDUs except those MDUs, which are in certain layer of users requirement, denoted as $K = H_{\varsigma}(C)(C$ is P, C or S, which means the projection of physical layer, structural layer or semantic layer$)$. According to the definition of MD, for each $M^D(M^D \in N)$, we have three kinds of $M^{D\prime}$. If we project to physical layer, the $M^{D\prime}$ should be $< M_p, \emptyset, \emptyset, N' >$. If we project to structural layer, the $M^{D\prime}$ should be $< \emptyset, M_c, \emptyset, N' >$. If we project to semantic layer, the $M^{D\prime}$ should be $< \emptyset, \emptyset, M_s, N' >$. Anyway, we can get $K(M^{D\prime} \in K)$ from H according to users requirement.

Additionally, it is possible that the input is not a MDU. For example, a user just wants to find some documents, which contain only some objects, from MDD. All what we need to do is to expend those objects to a MDU similarly in shape as $<MO, \emptyset, \emptyset>$. On the other hand, when the users requirement is to find some documents just contain certain multimedia document, we should take this MD apart to several MDU, and then inter-section the results. Obviously, it maybe lost some constrains especially in the mappings between MDUs. Thereby, make the results MDD larger than the actual results one. So, we should overload the ***contain*** function to exam the inclusive relation between MDUs again. It seems to what we discussed above, so we omit these detailed steps.

The example below give our an interpretation how to query data using the operations above.

Example 2. Find the picture which contain the *Golden Ball* from a MDD $M(1)$: $\{M^D{}_1, M^D{}_2, M^D{}_3, \ldots, M^D{}_e, \ldots, M^D{}_n\}$.

Step 0: expend Golden Ball to a MDU <{Golden Ball}, Ø, Ø >

Step 1: $M(1)_{\sigma}(<\{\text{Golden Ball}\}, \emptyset, \emptyset >)=\{ M^D{}_e \}$, denoted as $N(1)$

Step 2: $N(1)_{\varsigma}(<\{\text{Golden Ball}\}, \emptyset, \emptyset >)=\{<m_p, m_c, m_s, \{n(m_p, m_c), n(m_c, m_s)\}>\}$, denoted as $H(1)$

Step 3: $H(1)_{\pi}(P)=\{<<O_3, \emptyset, \emptyset>, \emptyset, \emptyset, \emptyset \}$, denoted as $K(1)$

Then, we can get the picture O_3, which contain the Golden Ball.

6 Conclusion

The kernel of the MIR system is to find the information user interested from a great deal of multimedia data. To achieve this purpose, the database of the MIR system must present the content of the multimedia data completely and strictly, but the present description methods of multimedia document, such as MPEG-7, have shortage, which cannot present the layered information. Aiming to this problem, according to the hierarchy category of abstraction of multimedia document's content, we propose a layered multimedia document description model and discuss the mapping relation between layers. Then, we define the formalized multimedia document using the model above. Lastly, we also explain how to operate this model. The future work will find more detailed operational ability of this model and design a proper query language.

Reference:

1. Johnson, R.B. Internet Multimedia Databases. In:IEEE Colloquium on Multimedia Databases and MPEG-7. 1999:1~6.
2. C. Barry, M. Lang. A Comparison of ‘Traditional’ and Multimedia Information Systems Development Practices. Information and Software Technology. 2003,45:217–227
3. P.H. Carstensen, L. Vogelsang, Design of Web-based information systems —new challenges for systems development?, Proceedings Ninth European Conference on Information Systems. 2001,June:27–29.
4. Carlo Meghini, Fabrizio Sebastiani and Umberto Straccia. A model of multimedia information retrieval. Journal of the ACM, Sept. 2001,48(5):909-970.
5. W.A.Khatib, Y.F.Day, A.Ghafoor, P.B.Berra, Semantic Modeling and Knowledge Representation in Multimedia Databases, IEEE Transaction on knowledge and data engineering, 1999,1/2:64-80
6. J. F. Yan, Y. Zhang, Z. H. Li, A Hierarchical Method to Describe Multimedia Document, International Conference on Computational Intelligence and Multimedia Applications (ICCIMA'03), IEEE-CS Press, To appear.
7. Faloutsos, C. Searching Multimedia Databases by Content. Kluwer Academic Publishers, Dordrecht, The Netherlands. 1996.
8. Faloutsos, C, Barber, R., Flickner, M., Hafner, J., and Niblick, W. Efficient and effective querying by image content. Journal of Intelligent Information Systems 3,1994(3/4):231–262.
9. Ozsu, M., Tamer. Issues in Multimedia Database Management. In: Proceedings of the International Symposium on Database Engineering and Applications (IDEAS’99), Montrea: IEEE-CS Press, 1999: 452~459.
10. Bach,J.R.,Fuller,C.,Gupta,A.,Hampapur,A.,Horowitz,B.,Humphrey,R.,Jain,R.,Andshu, C.-F.The Virage image search engine: An open frame work for image management. In Proceed ings of SPIE-96, 4th SPIE Conference on Storage and Retrieval for Still Images and Video Databases (San Jose, Calif.),1996:76-87.
11. Flickner, Myron et al., “Query by Image and Video Content: The QBIC System”, IEEE Computer, Volume 28, Number 9, September 1995, pp. 23-31.
12. W.W. Chu, A.F. Cardenas and R.K. Taira, “KMeD: a knowledge-based multimedia medical distributed database system,” Information Systems, Vol. 20, No. 2, 1995, pp.75-96.
13. E. Ozkarahan, “Multimedia document retrieval,” Information Processing & Management, Vol. 13, No. 1, 1995, pp. 113-131.
14. S. Marcus, V. S. Subrahmanian, Foundations of Multimedia Database Systems, Journal of the ACM, 43(3) pp 474-523, 1996.
15. V. Goebel, L. Eini, K. Lund, T. Plagemann, Design, Implenmentation, and Evaluation of TOOMM: A Temporal Object-oriented Multimedia Data Model, Preceeding of 8th IFIP 2.6 Working Conference, pp 145-168, January 1999.
16. Aiello,M.,Areces,C.,And Rijke,M. Spatial reasoning for image retrieval. In Proceedings of DL-99, 8th International Workshop on Description Logics (Linkoeping) SE, 1999:23-27.
17. Gudivada,V.N., And Raghavan,V.V. Design and evaluation of algorithms for image retrieval by spatial similarity. ACM Trans. Inf. Syst,1995,13(2)115-144.
18. A. B. Benitez, J. M. Martinez, H. Rising and P. Salembier, Description of a Single Multimedia Document, Book chapter in "Introduction to MPEG 7: Multimedia Content Description Language" edited by B. S. Manjunath, Phillipe Salembier and Thomas Sikora, Wiley, 2002.
19. J. F. Yan, Z. H. Li, W. J. Liu. Multimedia-Oriented Feature Link Model. Journal of Software , 2002,13(Suppl.): 205-212.