Learning rules for semantic video event annotation

Marco Bertini, Alberto Del Bimbo and Giuseppe Serra

Media Integration and Communication Center
University of Florence
Firenze, Italy
{bertini, delbimbo, serra}@dsi.unifi.it

Abstract. Automatic semantic annotation of video events has received a large attention from the scientific community in the latest years, since event recognition is an important task in many applications. Events can be defined by spatio-temporal relations and properties of objects and entities, that change over time; some events can be described by a set of patterns.

In this paper we present a framework for semantic video event annotation that exploits an ontology model, referred to as Pictorially Enriched Ontology, and ontology reasoning based on rules. The proposed ontology model includes: high-level concepts, concept properties and concept relations, used to define the semantic context of the examined domain; concept instances, with their visual descriptors, enrich the video semantic annotation. The ontology is defined using the Web Ontology Language (OWL) standard. Events are recognized using patterns defined using rules, that take into account high-level concepts and concept instances. In our approach we propose an adaptation of the First Order Inductive Learner (FOIL) technique to the Semantic Web Rule Language (SWRL) standard to learn rules. We validate our approach on the TRECVID 2005 broadcast news collection, to detect events related to airplanes, such as taxiing, flying, landing and taking off. The promising experimental performance demonstrates the effectiveness of the proposed framework.

1 Introduction and previous work

Video archives have grown steadily in the recent years. There is therefore the necessity to develop effective and efficient methods for automatic annotation and retrieval of information. Indexing of these archives, based on low-level visual features like color and texture, often does not meet the user's information needs due to the semantic gap between the information that can be extracted from the visual data and the interpretation of the same visual data by a user in a given context. Recently ontologies have been regarded as an appropriate tool to overcome this semantic gap. An ontology consists of concepts, concept proprieties, and their relationships and provides a common vocabulary that overcome semantic heterogeneity of information. Ontology Web Language (OWL) and Semantic Web Rule Language (SWRL) have been approved by W3C as language

standards for representing ontologies and performing reasoning using rules, respectively.
Recently several EC projects have addressed the problem of using ontologies for semantic annotation and retrieval by content from audio-visual digital libraries, among them AceMedia [1], Aim@Shape [2], Boemie [3] and VidiVideo [4].
Many researchers have built integrated system where the ontology provides the conceptual view of the domain at the schema level, and appropriate classifiers play the role of entities detectors. Once the observations are classified, the ontology is exploited to have an organized semantic annotation, establishing links between concepts and disambiguating the results of classification [5, 6].

Other researches have directly included in the ontology an explicit representation of the visual knowledge to perform reasoning not only at the schema level but also at the data level. Staab et al. [7] defined three separate ontologies that respectively modeled the application domain, the visual data and the abstract concepts, to perform the interpretation of video scenes. Automatically segmented image regions were modeled through low-level visual descriptors and associated to semantic concepts using manually labeled regions as training set. Kompatsiaris et al. [8] included in the ontology instances of visual objects that were used as references to perform the classification of the entities observed in video clips. They used as descriptors low-level perceptual features like color homogeneity, components distribution, and spatial relations. A similar solution was presented by Bertini et al. in [9], using generic and domain specific descriptors and introducing mechanisms for updating the prototypes of the visual concepts of the ontology, as new instances of visual concepts are added to the ontology; the prototypes are used to classify the events and objects observed in video sequences.
For event recognition several authors have exploited the ontology schema using temporal reasoning over objects and events. Snoek et al. [10] performed annotation of sport highlights using rules that exploited face detection results, superimposed captions, teletext and excited speech recognition, and Allen's logic to model temporal relations between the concepts in the ontology. Francois et al. [11] defined a special formal language to define ontologies of events and used Allen's logic to model the relations between the temporal intervals of elementary events, so as to be able to assess complex events in video surveillance. Haghi et al. [12] proposed to use temporal RDF to model temporal relationships in the ontology and provided examples of simple queries with temporal relationships between events. Bai et al. [13] applied temporal reasoning with temporal description logic to perform event annotation in soccer video, using a soccer ontology. All of these methods defined rules, used to describe events, that were created by human experts; thus, these approaches are not practical for the definition of a large set of actions.

To overcome this problem some researchers have studied techniques to learn automatically a set of rules. Dorado et al. [14] performed video annotation based on learned rules that infer high-level concepts from low-level features using decision tree technique. Shyu et al. [15] proposed a method to annotate rare events

and concepts based on set of rules that use low-level and middle-level features. Decision tree algorithm is applied to the rule learning process. Moreover they addressed the imbalance problem of positive and negative examples in the case of rare event/concept using data mining techniques. Liu et al. [16] proposed a method to enhance accuracy of semantic concepts detection, using association mining techniques to imply the presence of a concept from the co-occurrence of other high-level concepts. None of these three works is based on ontologies.
These methods that learn a set of rules by exploiting decision tree algorithms and low-level features, or simple junctions of high-level concepts, are not enough expressive to describe complex events. For example consider the event *A person enters in secured area.* This event can not be described using only the low-level descriptors of the person and of the area, or using the co-occurrence of the high-level concepts *person* and *secured area* since the person may stay outside of it, or may have always been inside it; instead it is required to take into account the temporal evolution of the characteristics and features of the objects and entities. This event can be fully described and modelled using first-order logic. A sentence that describes the events is: IF a person is outside of the secured area in the time interval t_1 AND the same person is in the secured area in the time interval t_2 AND t_1 is before t_2 THEN that person has entered the secured area; this sentence can be translated in the following fragment of first-order logic language:

$$\begin{aligned}&IF\ person(p)\ \wedge\ personOutsideOfSecuredArea(p, t_1)\ \wedge\\&personIsInSecuredArea(p, t_2) \wedge before(t_1, t_2)\\&THEN\ personEntersSecuredArea(p)\end{aligned}$$

where p is a variable that can be bound to any person and *t1* and *t2* are variables that are used to represent time intervals.

In this paper we propose a framework for video event annotation that exploits the Pictorially Enriched Ontology model, that includes concepts and their visual descriptors [9], and a method to learn sets of first-order logic rules that describe events defined in the ontology. Events can be described by spatio-temporal relations and properties of objects and entities, that change over time. The learned rules, defined using the SWRL, are applicable directly to an ontology defined using the OWL. The proposed learning method is an adaptation of the First Order Inductive Learner technique (FOIL [17]) to the Semantic Web technologies; for convenience this method will be referenced in the following as FOILS. This approach permits to create an ontology structure that allows to perform automatic semantic annotation of video sequences matching visual descriptors and recognizing events described using automatically learned rules. Moreover the learning approach used is more expressive than the previous methods because it defines rules through the first-order logic theory. To demonstrate the applicability to the automatic event annotation we show several events that can be recognized by automatically learned rules. In particular our tests are performed on the definition of some events related to airplane entities, defined in the LSCOM ontology.

2 Automatic video annotation framework

Our proposed framework, shown in Fig. 1, consists of three major components. In the shot segmentation and feature extraction component, the video is divided in syntactic units, low-levels features are extracted and objects and entities are identified. The Pictorially Enriched Ontology component includes a formal definition of a specific video domain and video structure. Rule-based reasoning is performed on high-level concepts and concepts instances for the automatic annotation of events. Rules are directly learned from the ontology using the FOILS algorithm.

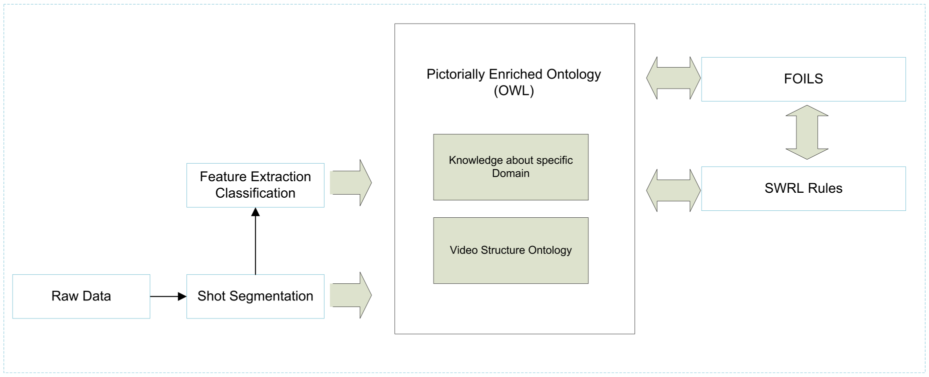

Fig. 1. Framework of the system

2.1 Video parsing and feature extraction

Video segmentation involves temporal partitioning of the video into units which serve as the basis for descriptor extraction and semantic annotation. In this work, shots are adopted as the basic syntactic unit, while video clips (video sequences possibly composed by more than one shot) are used as annotation units. For each shot visual descriptors such as color histograms, edge maps, etc. are extracted to perform a rough segmentation of each frame. Appropriate classifiers are applied to identify objects or entities. The feature extractors are used to provide the visual descriptors associated to the visual concepts of the ontology. These descriptors, that may be generic or domain specific, are then used to characterize concepts instances; this characterization allows to select the most representative concepts as visual prototypes of a concept, and allow to perform reasoning based on the visual appearance of a concept.

2.2 Pictorially Enriched Ontology

The Pictorially Enriched Ontology defines formally the domain of interest. In Fig. 2 is shown a simplified view of the main concepts used to represent the events related to airplanes, as studied in the use case. Video structure and visual descriptors associated to the visual concepts are stored in the ontology according to the features extraction and classifier detection results. The concept instances that are associated to visual descriptors can be used as matching references for the entities that have to be annotated. In our experiments the airplane concept is associated with color histograms, that are used by the tracker to identify the instances of the detected airplanes in a video sequence.

2.3 First-order rule learning

Terminology: To describe correctly the algorithm for learning sets of first-order rules, let us introduce some basic terminology from formal logic. All expressions are composed of constants (e.g. *Airplane1*, *Boeing-747*), variables (e.g. x, y), predicate symbols (e.g. *HasTrajectory*, *GreaterThan*) and function symbols (e.g. *duration*). The difference between predicates and functions is that predicates have value of *True* or *False*, whereas functions may have any constant as their value. In the following we will use lowercase for functions and capitalized symbols for predicates. A term is any constant, any variable, or any function applied to any term. A literal is any predicate or its negation applied to any term. If a literal contains a negation symbol ($\neg$), we call it *negative literal*, otherwise a *positive literal*. A *clause* is any disjunction of literals, where all variables are assumed to be universally quantified. A *Horn clause* is a clause containing at most one positive literal, as shown in the following:

$$H \vee \neg L_1 \vee \neg L_2 \ldots \vee \neg L_n$$

where H is the positive literal, and $\neg L_1 \vee \neg L_2 \ldots \vee \neg L_n$ are negative literals. It is equivalent to:

$$(L_1 \wedge L_2 \ldots \wedge L_n) \rightarrow H$$

which is equivalent to the following:

$$IF\ (L_1 \wedge L_2 \ldots \wedge L_n)\ THEN\ H$$

The Horn clause precondition $L_1 \wedge L_2 \ldots \wedge L_n$ is called *clause body*; the literal H that forms the post-condition is called the *clause head*.

First-Order Inductive Learner for SWRL technique: FOILS, first-order inductive learner for SWRL technique is an adaptation of the FOIL algorithm to the SWRL standard. The hypotheses learned by FOILS, similarly to FOIL, are sets of first-order rules, where each rule is similar to a Horn clause with the limitation that literals are not permitted to contain function symbols, in order to reduce the complexity of the hypothesis space search. At the beginning the algorithm starts with the *head* that we want to find in the rule and an empty or initial *body*. The algorithm iterates searching the new literals that have to be added to the body of the rule. The search is a general-to-specific search through the space of hypotheses, beginning with the most general preconditions possible (the empty or initial precondition), and adding literals one at a time to specialize the rule until it avoids all negative examples, or when no more negative examples are excluded for a certain number of loops. Two issues have to be addressed: the generation of hypothesis candidates and the choice of the most promising candidate.

Generating hypothesis candidates: Suppose that the current rule being considered is:

$$(L_1 \wedge L_2 \ldots \wedge L_n) \rightarrow P(x_1, x_2, \ldots, x_k)$$

where $(L_1 \wedge L_2 \ldots \wedge L_n)$ are literals forming the current rule preconditions and where $P(x_1, x_2, \ldots, x_k)$ is the literal that form the rule *head*. FOILS generates candidate specializations of this rule by considering new literals L_{n+1} that fit one of the following forms:

- $Q(v_1, \ldots, v_r)$ where Q is any predicate name occurring in *Predicates* and where the v_i are either a new variable or a variable already present in the rule. At least one of the v_i in the created literal must already exist as a variable in the rule.
- $Equal(x_j, x_k)$ where x_j and x_k are variables already present in the rule.

We observe that in FOIL there is another rule for generation of new candidates: it is the negation of either the above form of literals. This rule can not be exploited in our algorithm because it is not permitted by SWRL.

Most promising literal: To select the most promising literal from the candidates generated at each step, FOILS, similarly to FOIL, considers the performance of the rule over the training data. The evaluation function used to estimate the utility of adding a new literal is based on the number of positive and negative bindings covered before and after adding the new literal. More precisely consider some rule R, and a candidate literal L that might be added to the body of R. Let R' be the rule created by adding the literal L to rule R. The value of adding L to R is defined as:

$$Foil\_Gain(L, R) \equiv t\left(log_2 \frac{p_1}{p_1 + n_1} - log_2 \frac{p_0}{p_0 + n_0}\right)$$

where p_0 is the number of positive bindings of rule R, n_0 is the number of negative bindings of R, p_1 is the number of positive bindings of rule R' and n_1 is the number of negative bindings of R'. Finally, t is the number of positive binding of rule R that are still covered after adding literal L to R. When a new variable is introduced into R by adding L, then any original binding is considered to be covered as long as some binding extending it is present in the bindings of R'.

3 Use Case

We have applied the automatic video annotation framework to the detection of events related to airplanes, selecting them from the revised list of LSCOM events/activities [18]. Four events related to an airplane concept are analyzed: airplane flying, airplane takeoff, airplane landing, airplane taxiing. In Fig. 2 a simplified schematization of the ontology defined for these events is shown; for the sake of simplicity the visual descriptors associated to the airplane concept are not reported.

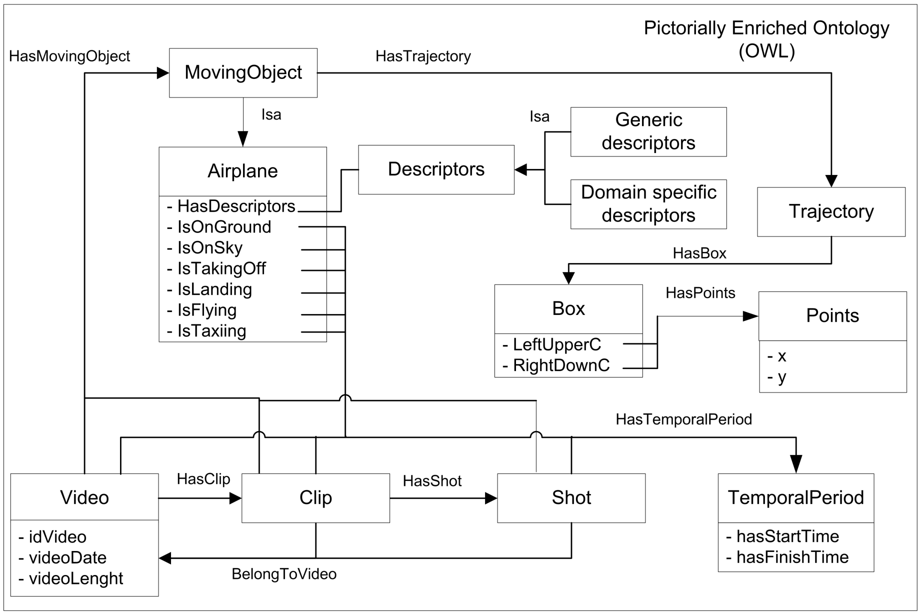

Fig. 2. Main concepts, relations and properties of the airplane events ontology.

These events can be detected using airplane, sky and ground detector and the temporal relationship between these concepts. For example, the evolution of an *airplane takeoff* event video is composed by a view of the airplane moving on the ground and after by a view of airplane on sky. An airplane detector has been created using the Viola&Jones object detector. The positive and negative examples used to train the detector have been selected from standard image datasets such as Caltech, VOC2005 and VOC2006. The negative examples used are images of man-made objects (e.g. other vehicles like cars, buses and motorcycles), outdoor scenes, animals and persons, various objects. The sky and ground detectors implemented are not used to classify all the parts and segments of each frame, but only locally, next to the airplane position, because it is enough to know if the airplane is on ground or in sky. The sky/ground detector evaluates statistical parameters of the luminance of the blobs around the detected airplane. Finally using a tracker, based on an improved version of the particle filter [19], we can determine the temporal evolution of the trajectory of airplane. The detected airplane, its bounding box trajectory and sky and ground detection are inserted in the ontology. Using learned SWRL rules the airplane takeoff, airplane landing, airplane flying and airplane taxiing events are identified. In Fig. 3 two examples of landing and take-off events are shown. These rules are learned from a set of positive and negative examples stored in the ontology using the FOILS technique, described in the previous section. To illustrate how the FOILS algorithm works we consider, for example, the target literal *AirplaneIsTakingOff*. The process starts with an initial rule written in SWRL. that models the take-off of airplane a within the video clip c:

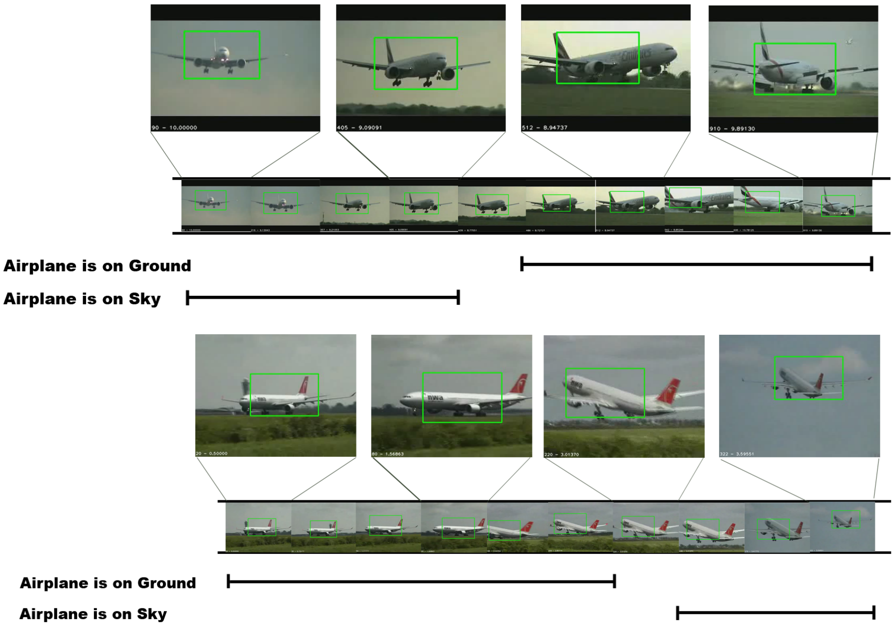

Fig. 3. Examples of airplane landing and take-off events. For each event the results of airplane detection and tracking are shown, along with a temporal model of the event.

$$Airplane(?a)\ \wedge Clip(?c)\ \rightarrow AirplaneIsTakingOff(?a,?c)$$

The initial candidates are all the classes and properties defined in the ontology domain and temporal properties used to encode Allen's logic. At each step the most promising literal is added, considering the performance of the rules over the training data.

N. detector	N. steps	Neg. examples	Pos. examples	Window size	Precision	Recall
1	17	3000	800	50×30	0.20	0.74
2	18	1500	800	50×30	0.19	0.83
3	20	1500	800	50×30	0.32	0.65
4	20	1500	800	25×10	0.75	0.55
5	22	1500	1040	50×30	0.41	0.66

Table 1. Precision and recall of airplane detector.

4 Experimental results

In the first part of the experiment we evaluate the performance of the airplane detector. We have trained five different detectors, using five configurations, with different numbers of positive and negative examples, image window sizes, and learning steps. Results are reported in Tab. 1. To train the fifth detector the number of positive examples of airplanes has been increased, adding more images of frontal and rear views of airplanes. The first three detectors did not provide an acceptable performance in terms of precision, as shown in the table. The decrease of the precision value between the fourth and fifth detector is mainly due to the fact that the detector may provide multiple detections for the same airplane, whose bounding boxes are overlapping, and these multiple detections have been counted as falses; without considering this overlapping effect the precision is comparable with that of the fourth detector. Considering this fact, the fifth detector has been selected and used in the following experiment.

To test the effectiveness of the learned rules we have used them to recognize events in a large dataset, that comprises 100 videos containing airplane events taken from the web[1] and 65 Trecvid 2005 videos.

The set of videos selected from the web video sharing sites (called on the following as Web Dataset) is available online, along with the airplane detector[2]. The Trecvid videos were selected from those reported in the LSCOM development set as containing the concepts *airplane_takeoff*, *airplane_landing* and *airplane_flying*, after a manual inspection that eliminated some errors of the ground truth (e.g. videos that contained rockets or helicopters instead of airplanes). Since the concept *airplane_taxiing* is not defined in LSCOM we inspected the videos annotated as containing *airplane* to select some videos that contained this event.

We have used an implementation of the FOILS algorithm, described in Sect. 2.3, to learn the SWRL rules that model the airplane events. The videos of the Web Dataset have been used to learn the rules. For each event that we want to learn we randomly select one third of the videos containing that event as positive examples, and one third of the videos of the other events as negative examples. In Tab. 2 the learned rules are shown. For each rule we present the initial rule and

[1] YouTube (http://www.youtube.com), Alice Video (http://dailymotion.alice.it), PlanesTV (http://www.planestv.com/planestv.html), Yahoo! Video (http://it.video.yahoo.com)

[2] http://www.micc.unifi.it/dome

Rule: Airplane TakingOff
Initial rule:
$Airplane(?p) \wedge Clip(?c) \rightarrow IsTakingOff(?p, ?c)$
Result rule:
$Airplane(?p)\ \wedge\ Clip(?c)\ \wedge\ IsOnSky(?p, ?g1)\ \wedge\ IsOnGround(?p, ?g2)\ \wedge$
$Temporal : after(?g1, ?g2)\ \wedge HasTemporalPeriod(?c, ?g3)\ \wedge Temporal : contains(?g3, ?g1)\ \wedge$
$Temporal : contains(?g3, ?g2)\ \wedge\ MovingObject(?p)\ \rightarrow\ IsTakingOff(?p, ?c)$

Rule: Airplane Landing
Initial rule:
$Airplane(?p) \wedge Clip(?c) \rightarrow IsLanding(?p, ?c)$
Result rule:
$Airplane(?p)\ \wedge\ Clip(?c)\ \wedge\ IsOnSky(?p, ?g1)\ \wedge\ IsOnGround(?p, ?g2)\ \wedge$
$Temporal : notafter(?g1, ?g2)\ \wedge HasTemporalPeriod(?c, ?g3)\ \wedge Temporal : contains(?g3, ?g1)\ \wedge$
$Temporal : contains(?g3, ?g2)\ \wedge\ MovingObject(?p)\ \rightarrow\ IsLanding(?p, ?c)$

Rule: Airplane Flying
Initial rule:
$Airplane(?p) \wedge Clip(?c) \rightarrow AirplaneFlying(?p, ?c)$
Result rule:
$Airplane(?p) \wedge Clip(?c) \wedge\ IsOnSky(?p, ?g1)\ \wedge$
$HasTemporalPeriod(?c, ?g2)\ \wedge Temporal : contains(?g2, ?g1)\ \rightarrow\ IsFlying(?p, ?c)$

Rule: Airplane Taxiing
Initial rule:
$Airplane(?p) \wedge Clip(?c) \rightarrow IsTaxiing(?p, ?c)$
Result rule:
$Airplane(?p)\ \wedge\ Clip(?c)\ \wedge\ IsOnGround(?p, ?g1)\ \wedge$
$HasTemporalPeriod(?c, ?g2)\ \wedge Temporal : contains(?g2, ?g1)\ \rightarrow\ IsTaxiing(?p, ?c)$

Table 2. Rules for airplane events recognition, obtained using FOILS.

the final rule obtained using FOILS. The learned rules recognize events within clips; this allows to cope with the case in which an event is shown using more than one shot. In some cases we can observe that FOILS learns some literals that are not necessary for the event definition, however this does not affect negatively the performance of the rule. This fact may happen since FOILS does not take into account the structure of the ontology; an example is the $MovingObject(?p)$ literal in the landing and taking-off rules, that is not necessary due to the fact that in our ontology this concept is an hypernym of airplane.

We have then applied the rules to the videos, evaluating the results, in term of precision and recall, for Web Dataset and Trecvid 2005 video separately and together, as shown in Tab. 3. As it can be observed the overall results for all the rules are extremely promising. Since the rules that describe flying and landing are more simple, their performance is better than that of the rules that model landing and taking-off. The main difference in the performance results between the two datasets is related to the quality of the images and to the presence of superimposed graphics, that were present only in the Trecvid news videos. Since the performance of the rules is dependent on the performance of the detectors and tracker we have investigated the cases in which the rules failed. The main cause of failure is due to the performance of the simple sky/ground detector, that uses only the luminance information. In a few cases the fault was the airplane detector, especially when superimposed graphics and text covered the appearance of the airplane.

Data Set	**Airplane Action**	**Precision**	**Recall**
Web Dataset	Airplane flying	0.96	0.94
Web Dataset	Airplane takeoff	0.76	0.80
Web Dataset	Airplane landing	0.84	0.96
Web Dataset	Airplane taxiing	1	0.76
TRECVID 2005	Airplane flying	0.94	0.5
TRECVID 2005	Airplane takeoff	0.3	0.42
TRECVID 2005	Airplane landing	0.66	0.66
TRECVID 2005	Airplane taxiing	1	0.76
Web Dataset + TRECVID 2005	Airplane flying	0.96	0.71
Web Dataset + TRECVID 2005	Airplane takeoff	0.61	0.70
Web Dataset + TRECVID 2005	Airplane landing	0.90	0.90
Web Dataset + TRECVID 2005	Airplane taxiing	0.94	0.84

Table 3. Precision and recall of Airplane flying, Airplane takeoff, Airplane landing, Airplane taxiing for different datasets

5 Conclusions

In this paper a framework for automatic event video annotation has been presented. A Pictorially Enriched Ontology has been defined, to perform automatic semantic annotation of videos, and a set of rules, used to describe events, has been learned from positive and negative video examples, using an adaptation of the First Order Inductive Learner technique to the Semantic Web Rule Language. The performance has been tested using different datasets to demonstrate the effectiveness of the proposed approach. Our future work will investigate techniques to incorporate learning of constants and function symbols in our framework, to permit to insert numerical temporal specifications in the event description.

Acknowledgment This work was partially supported by the Information Society Technologies (IST) Program of the European Commission VidiVideo Project (Contract FP6-045547).

References

1. aceMedia project (IST EC-FP6): Integrating knowledge, semantics and content for user-centered intelligent media services, http://www.acemedia.org/
2. Aim@Shape project (IST EC-FP6): Advanced and innovative models and tools for the development of semantic-based systems for handling, acquiring, and processing knowledge embedded in multidimensional digital objects, http://www.aimatshape.net/

3. Boemie project (IST EC-FP6): Knowledge acquisition from multimedia content, http://www.boemie.org/
4. VidiVideo project (IST EC-FP6): Improving the accessibility of video, http://www.vidivideo.info/
5. Snoek, C., Huurnink, B., Hollink, L., de Rijke, M., Schreiber, G., Worring, M.: Adding semantics to detectors for video retrieval. IEEE Transactions on Multimedia **9**(5) (August 2007) 975–986
6. Zha, Z.J., Mei, T., Wang, Z., Hua, X.S.: Building a comprehensive ontology to refine video concept detection. In: Proc. of ACM Int'l Workshop on Multimedia Information Retrieval, Augsburg, Germany (September 2007) 227–236
7. Simou, N., Saathoff, C., Dasiopoulou, S., Spyrou, E., Voisine, N., Tzouvaras, V., Kompatsiaris, I., Avrithis, Y., Staab, S.: An ontology infrastructure for multimedia reasoning. In: Proc. of VLBV, Italy (2005)
8. Dasiopoulou, S., Mezaris, V., Kompatsiaris, I., Papastathis, V.K., Strintzis, M.G.: Knowledge-assisted semantic video object detection. IEEE Trans. on Circuits and Systems for Video Technology **15**(10) (2005) 1210–1224
9. Bertini, M., Del Bimbo, A., Torniai, C., Cucchiara, R., Grana, C.: Dynamic pictorial ontologies for video digital libraries annotation. In: Proc. ACM Int'l Workshop on the Many Faces of Multimedia Semantics, Augsburg, Germany (2007) 47–56
10. Snoek, C., Worring, M.: Multimedia event-based video indexing multimedia event-based video indexing using time intervals. IEEE Transactions on Multimedia **7**(4) (2005) 638–647
11. Francois, A., Nevatia, R., Hobbs, J., Bolles, R., Smith, J.: VERL: an ontology framework for representing and annotating video events. IEEE Multimedia **12**(4) (Oct-Dec. 2005) 76–86
12. Qasemizadeh, B., Haghi, H., Kangavari, M.: A framework for temporal content modeling of video data using an ontological infrastructure. In: Proc. Semantics, Knowledge and Grid, Guilin, China (November 2006)
13. Bai, L., Lao, S., Jones, G., Smeaton, A.F.: Video semantic content analysis based on ontology. In: Proc. of Int'l Machine Vision and Image Processing Conference, Maynooth, Ireland (2007) 117–124
14. Dorado, A., Calic, J., Izquierdo, E.: A rule-based video annotation system. Circuits and Systems for Video Technology, IEEE Transactions on **14**(5) (May 2004) 622–633
15. Shyu, M.L., Xie, Z., Chen, M., Chen, S.C.: Video semantic event/concept detection using a subspace-based multimedia data mining framework. Multimedia, IEEE Transactions on **10**(2) (Feb. 2008) 252–259
16. Liu, K.H., Weng, M.F., Tseng, C.Y., Chuang, Y.Y., Chen, M.S.: Association and temporal rule mining for post-filtering of semantic concept detection in video. Multimedia, IEEE Transactions on **10**(2) (Feb. 2008) 240–251
17. Quinlan, J.R.: Learning logical definitions from relations. Mach. Learn. **5**(3) (1990) 239–266
18. Kennedy, L.: Revision of LSCOM event/activity annotations, DTO challenge workshop on large scale concept ontology for multimedia. Advent technical report #221-2006-7, Columbia University (December 2006)
19. Bagdanov, A.D., Del Bimbo, A., Dini, F., Nunziati, W.: Improving the robustness of particle filter-based visual trackers using online parameter adaptation. In: Proc. of IEEE Int'l Conference on AVSS, London, UK (September 2007) 218–223