Available online at www.sciencedirect.com
ScienceDirect
Information Sciences 178 (2008) 2534–2552
INFORMATION SCIENCES
AN INTERNATIONAL JOURNAL
www.elsevier.com/locate/ins

Natural language querying for video databases

Guzen Erozel [a], Nihan Kesim Cicekli [a], Ilyas Cicekli [b,*]

[a] Department of Computer Engineering, METU, Ankara, Turkey
[b] Department of Computer Engineering, Bilkent University, Ankara, Turkey

Received 13 November 2006; received in revised form 31 January 2008; accepted 6 February 2008

Abstract

The video databases have become popular in various areas due to the recent advances in technology. Video archive systems need user-friendly interfaces to retrieve video frames. In this paper, a user interface based on natural language processing (NLP) to a video database system is described. The video database is based on a content-based spatio-temporal video data model. The data model is focused on the semantic content which includes objects, activities, and spatial properties of objects. Spatio-temporal relationships between video objects and also trajectories of moving objects can be queried with this data model. In this video database system, a natural language interface enables flexible querying. The queries, which are given as English sentences, are parsed using link parser. The semantic representations of the queries are extracted from their syntactic structures using information extraction techniques. The extracted semantic representations are used to call the related parts of the underlying video database system to return the results of the queries. Not only exact matches but similar objects and activities are also returned from the database with the help of the conceptual ontology module. This module is implemented using a distance-based method of semantic similarity search on the semantic domain-independent ontology, WordNet.

Keywords: Natural language querying; Content-based querying in video databases; Link parser; Information extraction; Conceptual ontology

1. Introduction

The current technological developments offer convenient ways for storing and querying video files that include movies, news clips, sports events, medical scenes, security camera recordings, to people and researchers working in the areas of media, sports, education, health, security, and many others. There are mainly two problems in the implementation of video archives. The first problem is related to the modeling and storage of videos and their metadata. The second problem is how to query the content of the videos in a detailed and easy manner.

* Corresponding author. Tel.: +90 312 2901589; fax: +90 312 2664047.
E-mail addresses: guzen.erozel@tcmb.gov.tr (G. Erozel), nihan@ceng.metu.edu.tr (N.K. Cicekli), ilyas@cs.bilkent.edu.tr (I. Cicekli).

0020-0255/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2008.02.001

Unlike relational databases, spatio-temporal properties and rich set of semantic structures make it more complex to query and index the video content. Due to the complexity of video data, there have been many video data models proposed for video databases [1,2,8,10,15,16,22–24]. Some of the existing work use annotation based modeling. Some use physical level video segmentation approach [37,44], and some have developed object based modeling approaches which use objects and events as a basis for modeling the semantic information in video clips [1,30]. The object-oriented approach is more suitable to model the semantic content of videos in a more comprehensive way.

There have been several methods proposed to query the content of video databases in the literature. It is possible to divide these methods into mainly two groups: graphical interfaces and textual interfaces. In the graphical user interfaces, the user generates queries by selecting proper menu items, sketching graphs, drawing trajectories and entering necessary information with the help of a mouse like in WebSEEK [25], SWIM [43] and VideoQ [7]. These are in general easy to use systems but they are not flexible enough. On the other hand, textual interfaces that require the user to enter queries via SQL-like query languages or extensions to SQL are difficult to use, since the user has to learn the syntax of the language [10,26]. Other approaches for textual interfaces are not so flexible for the reason that Boolean operators or category-hierarchy structures are used for querying like in VideoSTAR [17] and VISION [27]. The most flexible method among all these approaches is the use of a natural language.

The aim of this paper is to present a natural language query interface over a content-based video data model which has spatio-temporal querying capabilities in addition to the basic semantic content querying. The video data model [22] identifies spatial properties of objects with rectangular areas (regions) resembling MBRs (minimum bounding rectangles). It is possible to compute and query spatial relationships between two rectangular areas, hence the objects covered by those rectangles. It is also possible to handle spatial relations such as left, right, top, bottom, top-left, top-right, bottom-left, bottom-right, as directional relations, and overlaps, equal, inside, contain, touch, and disjoint as topological relations. The model also supports querying the trajectory of an object given the starting and ending regions. The model allows us to perform spatio-temporal queries on the video and also provide the inclusion of fuzziness in spatial and spatio-temporal queries. Our video data model [22] previously had only a graphical query interface. Later, the system is integrated with a natural language interface in which the user can express queries in English. In this paper, we present this natural language query interface to the video database system. The capability of querying the system in a natural language instead of an artificial language can be exemplified with the following kinds of queries.

- *Find the frames where the prime minister meets the minister of foreign affairs.* (A journalist may be posing this kind of query frequently.)
- *Show all intervals where the goals are scored.* (This query may be used in a sports event archive.)
- *Show all cars leaving the parking lot.* (A security camera recording can be queried in this fashion.)

Our natural language interface can handle such queries and other forms of queries given in English.

There has been a considerable amount of work in querying the video frames in natural languages. They use syntactic parsers to convert the media descriptions (or annotations) and build semantic ontology trees from the parsed query [29]. However, these are usually application specific and domain-dependent (e.g. querying only the recordings of street cameras in SPOT [19] or querying only the parts of news broadcast in Informedia [18]). Not every system using natural language can capture high-level semantics. The video system Informedia which is using keyword-matching natural language interface, cannot answer detailed queries nor handle structures with attributes. In this paper we propose a general-purpose video database querying system by adding a natural language interface to a video data model [22]. Another contribution of the querying facility of the system is the usage of information extraction techniques in order to find the semantic representation of user queries [12,13]. In SOCIS system, the crime scene photographs are annotated with text and keywords are extracted to index the photos [11,31]. However, only spatial relations in images are extracted in that system. In our system, on the other hand, many other query types can be extracted from sentences and their semantic representations are mapped to the underlying video data model.

It is an important problem to match a given query with the underlying video data in the systems that use natural language interfaces. When natural language queries are parsed, the first aim is to extract the entities

that occur in the query and match them with entities in the database. However sometimes, an exact match cannot be obtained for the query from the database. For example, the user may query a *car* where a *car* entity does not exist but instead *Mercedes* and *Fiat* exist as video entities. In order not to reply with an empty result set, ontology-based querying is used after the parsing phase. The similarity between entities in the database and parsed entities from query is evaluated by using an *is-a* hierarchy. The root of the tree is semantically more generalized than the leaves. The highest similarity value of the entity is selected to be in the result. Therefore, a natural language interface that uses ontology-based querying returns close-match results in addition to exact matches [3,21].

Another important contribution of the natural language query interface presented in this paper is to perform an ontological search by using a domain-independent general-purpose ontology that holds the ontological structures of English words. In querying, the system will not only search the given words but also perform semantic similarity search based on the ontological structure of the given words. For instance, when the user poses a query like "*Show all frames where vehicles are seen*", the system will be able to return videos which include *vehicles* and all semantically similar words such as *cars*, *buses* or *trains*. Although many different semantic similarity algorithms exist, none of the methods gives the best result. In our system, we preferred a combined method of an edge counting method of Wu and Palmer [41] with a corpus based method, because it gave the best results in our tests [12]. The ontology-based querying has previously been used in some other video systems, but these systems construct their own ontology that needs to be changed whenever the domain changes [29]. In this paper, syntactic parsing with a general lexicon and domain-independent ontology search are preferred for the flexibility of the developed interface. Hence, no additional dictionary or semantic similarity algorithm is needed when the domain and the video data entities change.

The rest of the paper is organized as follows: The related work is given in Section 2. In Section 3, the video data model that is used as the basis of this paper is summarized by introducing the types of queries supported by the model. The proposed system maps the given English queries into their semantic representations. The semantic representations are built from the output of the parsing module, by the information extraction module of the system. Section 4 describes the extraction module and the parsing technique used in query processing. Section 5 presents the details of the ontology-based querying that provides close-match results. In Section 5, we also explain the expansion of the semantic representations that are extracted from the natural language processing module. We give evaluation results for ontology-based searching in Section 6. Finally, Section 7 presents the conclusions and future work.

2. Related work on natural language query processing

A natural language interface is desirable to query the content of videos, in order to provide a flexible system where the user can use his/her own sentences for querying. The user does not have to learn an artificial query language, which is a major advantage of using a natural language in querying.

Although natural language interfaces provide the most flexible way of expressing queries over complex data models, they are limited by the domain and by the capabilities of parsers. The main issue is the conversion of a given natural language query into the semantic representation of the underlying query language. This process is not a simple task and different NLP techniques can be employed in order to map queries into their semantic representations. Before we describe NLP techniques that are used in our system, we review the related work in this section.

2.1. Natural language querying over databases

Early studies of natural language query processing depend on simple pattern-matching techniques. These are simple methods that do not need any parsing algorithm. SAVVY [5] is an example of this approach. In this system, some patterns are written for different types of queries and these patterns are executed after the queries are entered. For example, consider a table consisting of country names and their capitals. Suppose that a pattern is written as "*Retrieve the capital of the country if the query contains the word 'capital' before a country name*". Then the query "*What is the capital of Italy?*" will answer "*Rome*" as the result. However, since the results of this technique were not satisfactory, more flexible and complex techniques have been investigated.

The method used in the system LUNAR [39] supports a syntax-based approach where a parsing algorithm is used to generate a parse tree depending on user's queries. This method is especially used in application-specific database systems. A database query language must be provided by the system to enable the mapping from parse tree to the database query. Moreover, it is difficult to decide the mapping rules from the parse tree to the query language (e.g. SQL) that the database uses.

The system LADDER [5] uses semantic grammars where syntactic processing techniques and semantic processing techniques are used together. The disadvantage of this method is that semantic approach needs a specific knowledge domain, and it is quite difficult to adapt the system to another domain. In fact, a new grammar has to be developed when the system is configured for a different domain.

Some intermediate representation languages can be used to convert the statements in natural language to a known formal query language. MASQUE/SQL [4] is an example for this approach. It is a front-end language for relational databases that can be reached through SQL. User defines the types of the domain which database refers using an *is-a* hierarchy in a built-in domain-editor. Moreover, words expected to appear in queries with their logical predicates are also declared by the user. Queries are first transformed into a Prolog-like language LQL, then into SQL. The advantage of this technique is that the system generating the logic queries is independent from the database and therefore, it is very flexible in domain replacements.

2.2. Natural language techniques over video databases

Because of rich set of semantic structures and spatio-temporal properties in video data models, it is more complex to support querying in video databases. Natural language querying systems should be able to handle more complex query structures. This means that NLP techniques used in video databases should be more sophisticated so that queries can be mapped into the underlying query language. Syntactic parsers can be used to parse the given natural language queries, mapping systems can be used to map queries into their semantic representations, and ontologies can be used to extend the semantic representations of the queries.

The video system, SPOT [19], can query moving objects in surveillance videos. It uses the natural language understanding in the form of START (a question-answering system) [20], which has an annotation based natural language technique. Annotations which are English phrases or sentences, are stored in the knowledge base. The phrases are used to describe question types and information segments. Queries are syntactically parsed to match with these annotations. When a match is found between annotations and parsed query phrases, the segment in the annotation is shown to the user as a result. In SPOT, a track is the basic unit of data, which traces the motion of a single object in time. The queries are translated into symbolic representations that tracks are formulated. These representations are also in the sense of matching the annotations in the knowledge base. However, this system is incapable of capturing high-level semantics of the video content.

In [29], media data in the query is extracted into *description data* by using a matcher tool that uses the lexicon. These descriptions are semantically parsed with a domain specific lexicon in order to be matched with the data in the database. This method is used for exact matching. However, since in natural language, same descriptions can have different semantics, approximate matching is performed in the system by using semantic network model. When the query is parsed, the semantic representation is translated into a semantic network in which nouns and the actions in the query are the major nodes. There are also domain-dependent verb and noun hierarchies stored in the knowledge base of the system. The semantic networks are tried to be matched node by node according to the hierarchies. Weights are used in the hierarchy trees to enable better matching results. The main drawbacks of the system are the difficulty of the weights for approximate matches and the usage of a domain-dependent lexicon.

Informedia [18] uses a natural language interface in searching a news-on-demand collection. When a user poses a natural language query, the system searches and retrieves the best 24 news stories that match the query. Text summary headlines appear for the selected stories and the user can select the story that he is most interested. Both text headlines and video "skims" are generated by extractive summarization. All stories are scanned for words that have a high inverse document frequency in order to determine distinguishing stories. The major concern in this study is to search the text annotations by using summarization techniques. They mainly use a keyword-based search engine to find the video clips. Their approach is quite different from ours, since we store the video content using a video data model not as text annotations.

VideQA [42] uses a natural language interface in a question-answering system on news videos. In VideQA, a short question is mapped into one of the eight general question classes with a rule-based question classifier. In order to cope with the imprecise information in short questions, they also use WordNet to expand short questions.

In [14], a natural language query processor based on conceptual analysis is described. Their conceptual analyzer first tries to find nouns in the given query and then tries to fill the templates induced by verbs with the found nouns. A filled template represents the semantic structure of a given natural language query. They use a domain-dependent lexicon for nouns and verbs.

BilVideo [10] is expanded to support a natural language interface [23]. Natural language queries are mapped into Prolog-based fact representations. Since this system does not support ontology-based querying, it is not possible to get close-match results.

Zhang and Nunamaker use natural language processing techniques in video indexing [43]. They use a natural language approach to a content-based video indexing to find video clips. Their technique is similar to information retrieval techniques. They did not use any ontology to find close-match results.

The natural language query interface described in this paper uses a wide-coverage shallow parser for English to parse the given queries. Then, the information extraction module is used to map the parsed queries into the underlying formal query forms. Since the coverage of the parser and the information extraction module is high, the system can handle a lot of different query forms. These two steps can be seen as first figuring out the question template from the query and then filling in this template. Later, the found template is expanded using WordNet which is a wide-coverage domain-independent ontology in order to handle approximate matches.

Domain dependence is an important subject to consider for every natural language interface method. Domain dependency should be kept to minimum in order to enable a more flexible system when deciding on a technique to implement a natural language interface. The systems that use domain-dependent ontology only, need different ontologies for databases in different domains. In our system, however, we use WordNet which is a wide-coverage domain-independent ontology for English in order to provide maximum flexibility in querying.

3. Video data model

The video data model used in this paper is a content-based spatio-temporal video data model [22]. The basic elements of the data model are *objects*, *activities* and *events*. The video clip is divided into time-based partitions called *frames*. Objects are real world entities in these frames (e.g. book, Elton John, football, etc.). They can have properties (or attributes) like name and quantifiers (size, age, color, etc.). Activities are verbal clauses like playing football, singing, etc. Events are detailed activities that are constructed from an activity name and one or more objects with some roles. For instance *John plays football*, and *the cat is catching a mouse* are events.

Frame sequences contain a set of continuous frames that include any semantic entity like an object, an event, etc. Each entity in the video data model is associated with a set of frame sequences in which they occur. Frame contents can be queried by giving the object/event or activity of interest, and the system will return the relevant frames with the display option.

The basic type of queries supported in our system is occurrence queries. Occurrence queries are used to retrieve frames in which a given object, event or an activity occurs. In addition, it is possible to retrieve objects, events and activities occurring in a given frame (time) interval.

The support for spatio-temporal queries is the main concern in this video data model. Spatial properties contain the location of an object in the video frame. It is more difficult to represent the spatial relationships between two objects in a video than images since video data has time-dependent properties. In this model, a two-dimensional coordinate system is used for spatial properties of objects. The most preferred method to define the location of an object is to use an MBR (minimum bounding rectangle) which is an imaginary rectangle that bounds the area of an object at the minimum level. With the spatial properties added to the model, temporal properties are combined with them by defining region-interval tuples for objects. Thus, it is possible to define and query spatio-temporal relationships in a frame sequence between any two objects. A rule base covering the relations *top, bottom, right, left, top-right, top-left, bottom-right, bottom-left*, etc. is defined to help calculations of spatial relationships. Since the objects may move in a given interval, the spatial relationships may change over time. For instance, the *cat is to the left of the table* may change in a given time interval. This problem introduces fuzzy definitions of spatial relations in the model [22].

Fig. 1. Graphical query interface of the video database system.

The relative positions of two objects or an object's own position in the frame can be queried using spatial relations. The spatial relations between objects can be fuzzy since the objects may be moving in a video stream. The data model incorporates fuzziness in the querying of spatial relations by introducing a threshold value in their definition. Temporal properties are given as time intervals described in seconds and minutes. In the implementation, spatial queries are called regional queries; temporal queries are called interval queries. It is also possible to query the trajectory of a moving object. Starting from one region, an object's trajectory can be queried if its positions are adjacent up to an ending region in consecutive video frames.

A graphical user interface is used to query the videos in a previous implementation of the video database system. Pull-down menus and buttons are used to select objects, events, activities and spatial relations to express a query as seen in Fig. 1. When a spatial relation is queried, related objects, the spatial relation and also a threshold value are chosen from the drop down lists, and the type of the query must be selected using buttons. However, this interface has not been very flexible. Therefore, we have decided to use a natural language interface for querying the video contents.

4. Query processing

The idea is to map English queries into their semantic representations by using a parser and an information extraction module. The semantic representations of queries are fed into the underlying video database system to process the query and show the results. The main structure of the system is given in Fig. 2.

4.1. Semantic representations of queries

In order to extract the semantic representation of a query, it is sufficient to find the type of the given query and its parameters. The structure of the semantic representations is similar to the underlying data model structures. Therefore, in order to obtain the semantic representation of a query, we should be able to determine which parts of the query determines the type of the query and which parts correspond to the parameters of the query.

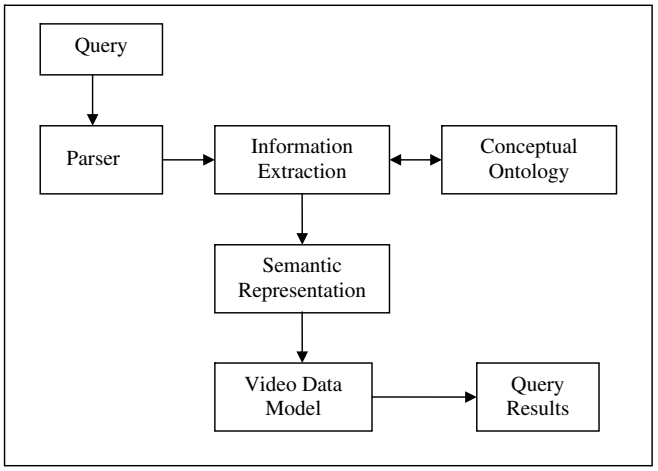

Fig. 2. The main structure of the natural language interface of the video data model.

Every query should include at least an object or an activity. Objects and activities are atomic particles that form an event. Objects can also have parameters like its name and attributes that qualify its name in the query, such as color, size, etc. In the implementation, the object representation is restricted to have only two attributes described by any adjectives in the query. Therefore, the atomic representation of an object is:

- *Object(name, attribute1, attribute2)*

where 'Object' is the predicate name used in the semantic representation of the query, 'name' is the name of the object, attributes (if they exist) are the adjectives used to describe the object in the query.

Activities are just verbs that are focused in the video frames. So they are also atomic and have representations like:

- *Activity(activity_name)*

where *Activity* is the predicate name used in the semantic representation of the query, *activity_name* is the activity verb itself.

Events are not atomic, because every event has an activity and the actors of that activity as its parameters. Thus, an event will be represented as:

- *Event (activity, object1, object2...)*

where *Event* is the predicate name, *activity* is the activity of this event, and the subsequent objects are the actors of this activity. The full semantic representation of an event occurrence query can be constructed only after the activity and objects are extracted.

There may be other kinds of semantic representations for spatial and temporal properties in the queries. Some of them are atomic structures such as coordinates and minutes, and some of them are relations between any two object entities. Regional queries include some rectangle coordinates to describe a region. During information extraction, the phrases representing the rectangles are converted to two-dimensional coordinates. Thus, regional semantic representation is:

- *Region(x1, y1, x2, y2)*

where *Region* is the predicate name that is used in the query semantic representation. $x1$ and $y1$ are the coordinates of the upper-left corner; $x2$ and $y2$ are the coordinates of the lower right corner of MBR.

Temporal properties are encountered as intervals in the query, so an interval is represented as follows:

- *Interval(start, end)*

where *start* and *end* are the bounding frames of the interval.

Spatial relations are extracted as predicates and the extracted objects involved in the spatial relations become the parameters of the predicates in the semantic representations. Semantic representations of the supported spatial relations are:

- *ABOVE (object1, object2, threshold)*
- *RIGHT (object1, object2, threshold)*
- *BELOW (object1, object2, threshold)*
- *LEFT (object1, object2, threshold)*
- *UPPER-LEFT (object1, object2, threshold)*
- *UPPER-RIGHT (object1, object2, threshold)*
- *LOWER-LEFT (object1, object2, threshold)*
- *LOWER-RIGHT (object1, object2, threshold)*

In these predicates, *threshold* value is used to specify the fuzziness in the spatial relations. The threshold value is between 0 and 1, and it indicates the acceptable correctness percentage for the relation. For example, the query "Find the frames, in which object A is seen to the left of object B, with a threshold value of 0.7" is a fuzzy spatial relationship query. The system finds the frames in which A and B occurs and regions of A and B satisfies the spatial relationship *LEFT* with at least 70% correctness, and these frames are returned as the result of this fuzzy query.

Supported query types and their semantic representations are given in Table 1. Each query in Table 1 has a different semantic representation with a different set of parameters. Hence, the data to be extracted depend on the type of the query. The semantic representations of the parameters are extracted, and they are combined to get the final semantic representation of the query.

4.2. Parsing queries

A syntactic parser is needed to extract information from the user query. We only need specific kinds of word groups (like objects, activities, start of the interval, etc.) to obtain the semantic representations. A light-parsing algorithm such as shallow parser [38], chunk parser [33] and link parser [28,36] is enough for our purposes, since there is no need to find the whole detailed parse tree of a query. We have chosen to use a link parser to parse given queries in our implementation, because of its ability to give more accurate results. Another advantage of this parser is to have the ability to get the grammatical relations between word groups, and these relations are used in the extraction of semantic representations.

Link parser is a kind of light parser which parses one English sentence at a time [28,36]. When a sentence is given as an input to the parser, the sentence is parsed with *linkages* using its grammar and its own word dictionary. As described in [28,36], a link grammar links every word in the sentence. A link is a unit that connects two different words. The sentence can be described as a tokenized input string by links which are obtained by the sentence splitter. When the sentence is parsed, it is tokenized with linkages (a group of links that do not cross). In the following example, Ds is a link that connects the singular determiner with its noun.

```
+---Ds---+
a        cat
```

A determiner (here it is a) must satisfy a Ds connector to its right. A single noun (here it is cat) must satisfy a Ds connector to its left. When the connectors are plugged, a link is drawn between a word pair.

When a natural language query is parsed by the link parser, the output of the parser includes the linkage information between the words of the query. For instance in Fig. 3 no two links cross and no words are left as unlinked. The links are represented in capital letters. The semantics of the links are as follows:

O	connects transitive verbs to objects
D	connects determiners to nouns
M	connects nouns to post-nominal modifiers

Table 1
Query types supported by the system, semantic representations and their examples

Query types	Semantic representations of queries	Query examples in natural language	Semantic representation of the examples
Elementary object queries	RetrieveObj (objA): *frame_list*	Retrieve all frames in which John is seen	• RetrieveObj (Obj_A): *frames.* • Obj_A (John, NULL, NULL)
Elementary activity type queries	RetrieveAct (actA): *frame_list*	Find all frames in which somebody plays football	• RetrieveAct (Act_A): *frames* • Act_A (play football)
Elementary event queries	RetrieveEvnt (evtA): *frame_list*	Show all frames in which Albert kills a policeman	• RetrieveEvnt (Evnt_A): *frames* • Evnt_A (Act_A, Obj_A, Obj_B) • Act_A (kill) • Obj_A (Albert, NULL, NULL) • Obj_B (policeman, NULL, NULL)
Object occurrence queries	RetrieveIntObj (intervalA): *object_list*	Show all objects present in the last 5 min in the clip	• RetrieveIntObj (Int_A): *objects* • Int_A($x - 5$, x). [x: Temporal length of video]
Activity type occurence queries	RetrieveIntAct (intervalA): *activity_list*	Retrieve activities performed in the first 20 min	• RetrieveIntAct (Int_A): *activities* • -Int_A (0, 20)
Event occurrence queries	RetrieveIntEvt (intervalA): *events_list*	Find all events performed in the last 10 min	• RetrieveIntEvt(Int_A): *events* • Int_A($x - 10$, x). [x: Temporal length of video]
Fuzzy spatial relationship queries	RetrieveObj_ObjRel (rel,threshold): *frame_list*	Find all frames in which Al Gore is at the left of the piano with the threshold value of 0.7	• RetrieveObj_ObjRel (LEFT, 0.7): *frames* • LEFT (Obj_A, Obj_B) • Obj_A (Al Gore, NULL, NULL) • Obj_B (piano, NULL, NULL)
Object interval queries	RetrieveIntervalofObj (objA): *interval_list*	When is Mel Gibson seen?	• RetrieveIntervalofObj (Obj_A): *intervals* • Obj_A (Mel Gibson, NULL, NULL)
Activity interval queries	RetrieveIntervalofAct (actA): *interval_list*	Retrieve intervals where somebody runs	• RetrieveIntervalofAct (Act_A): *intervals* • Act_A (run)
Event interval queries	RetrieveIntervalofEvnt (evtA): *interval_list*	Find all intervals where the cat is running	• RetrieveIntervalofEvnt(Evnt_A): *intervals* • Act_A (run) • Evnt_A (Act_A, Obj_A, NULL) • Obj_A (cat, NULL, NULL)
Regional(frame) queries	RetrieveObjReg (objA, region): *frame_list*	Show all frames where Bill is seen at the upper-left of the screen	• RetrieveObjReg (Obj_A, Reg_A): *frames* • Obj_A (ball) • Reg_A($x/2, 0, x, y$) [if coordinates of the frame's rectangle is considered as 0, 0, x, y]
Regional(interval) queries	RetrieveObjInt (objA, intervalA): *region_list*	Find the regions where the ball is seen during the last 10 min	• RetrieveObjInt (Obj_A, Int_A): *regions* • Obj_A (ball, NULL, NULL) • Int_A (Int_A($x - 10$, x) [x: Temporal length of video]
Trajectory queries	TrajectoryReg(objA, start_region, end_region): *frame_sequence*	Show the trajectory of a ball that moves from the left to the center	• TrajectoryReg (Obj_A, Reg_A, Reg_B): *frames* • Obj_A (ball, NULL, NULL) • Reg_A (0, 0, $x/2$, y) • Reg_B($x/4$, $y/4$, $3x/4$, $3y/4$) [if coordinates of the frame's rectangle is considered as 0, 0, x, y]

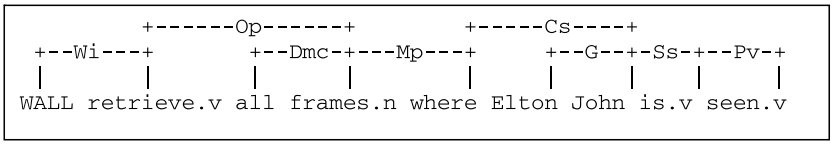

Fig. 3. The output of link parser for a sample object query.

C	connects subordinating conjunctions
G	connects proper nouns together in series
S	connects subject-nouns to finite verbs
P	connects forms of the verb "be" to various words
Wi	connects imperatives to the *wall* (beginning of the sentence)

The lowercase letters next to the representation of link types are the attributes of the links. For instance p means "plural" in Op, mc means "plural countable" in Dmc, s means "singular" in Cs, and v means "passive" in Pv.

4.3. Information extraction module

After a query is parsed with the link parser, the information extraction module forms the semantic representation of the query from the output of the parser. A similar technique is also used in crime scene reconstruction [11] which has been adopted from information extraction methodology used in SOCIS [31]. In [11], crime photos are indexed with spatial relations and scene descriptions by using the information extraction in an application domain. In our system, the information extraction module depends on a rule-based extraction defined on the linkages of a link parser. The aim is to extract word groups in the parsed query in order to map them to the semantic representations defined for *objects*, *activities, intervals, regions* and *spatial relations*. After extracting the basic terms, the full representation of the query is formed by determining the query type and the return value. Objects are nouns and their attributes are adjectives; activities are verbs, regions and spatial relations can be nouns or adjectives. So, the link types and the order of the links determine what it is to be extracted.

For each word group that can be extracted, one or more rules are implemented in a knowledge base. Once the query is parsed, special link types are scanned. Whenever a special linkage path is found, the rules written for finding out the structure (like object, query type, event, etc.) are applied to the path. For example, the following rule is one of the rules that are used to extract an activity:

- *Control the Cs link.*
- *If an Ss link follows this link and if the right-end of Cs is any word like "somebody, anybody, someone, etc.". Ss link's right word is the activity.*
- *If there is a following Os link, then the right-end of Os is a part of the activity (ex: playing football)*

For each query, the query type is first extracted from the parsed query. Then, the parts of the semantic representation are extracted. For instance, consider the query in Fig. 4. In this query, the word `interval` is the

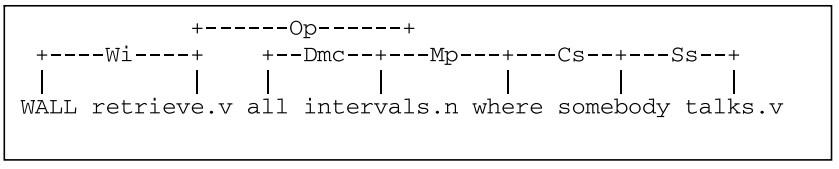

Fig. 4. The output of link parser for a sample interval query.

key word to determine the return value of the query's semantic representation, and it indicates that the query is an interval query. In order to determine the type of this interval query (i.e. *object interval query*, *activity interval query*, or *event interval query*), the right-end of `Mp` link is analyzed. For instance, when the rule above is applied, it is determined that there is an activity in the query. Therefore, this is an *activity interval query*, and the activity is *talk*. Thus, the following semantic representation is found by the information extraction module after all relevant rules are applied:

- RetrieveIntervalofAct (Act_A): *intervals*.
- Act_A (talk).

Let us consider the query in Fig. 3 as another example. `Op` linkage helps us to determine the return type of the query as *frames*. Then all atomic representations are searched tracing the linkage paths. When the `Cs` link is traced, an object, *Elton John*, is found. The tracing process is finished when no more atomic representations can be found. So depending on the keyword *frames* and only one object representation, the query type is decided to be an *Elementary Object Query*. Thus, we get the following semantic representation for the query.

- RetrieveObj (Obj_A): *frames*.
- Obj_A (Elton John, NULL, NULL).

Certain parts of the parsed query may not directly map to a part of the semantic representation. For instance, a numerical value can be entered either as a number or as a word phrase in a query. However in the data model it needs to be a numerical value. Therefore, a numerical value expressed as a word phrase should be converted into a number. This difficulty also arises in the extraction of regions. The regions are preferred to be described as areas relative to the screen like *left, center, upper-left*, etc. To map this data to the video data model, these areas should be converted into two-dimensional coordinates. Thus, the regions are represented as rectangles. So, the screen is assumed to be divided into five regions as *upper-right*, *upper-left*, *lower-left*, *lower-right* and *center*. The area in the query is matched with these regions. For example, for the word phrase "*right*" in the query, the coordinates of upper-right and lower-right are evaluated. A similar problem also occurs in interval queries. When the user phrases "the *last 10 min*", the beginning time must be evaluated to map with the video data. Therefore, the extraction algorithm is also responsible for these conversions. Some examples of these conversions are given in the last column of Table 1.

5. Ontology-based querying

In our video data model, all objects in the video database are annotated with nouns, and all activities are annotated with verbs. When a natural language query is converted into its semantic representation, the semantic representation of the query will contain at least one object or one activity. Similarly, each query object is represented with a noun and each query activity is represented with a verb. In order to find the video frames in which the query object appears, the noun representing the query object must match with one of the nouns representing the objects in the video frames. When the query object exactly matches a video object, we call this as an exact match.

When there does not exist any exact match between the object in the semantic representation and any of the objects in the video database, the system cannot return any results if it employs only an exact match method. This problem occurs when the user enters more generalized or more specific words in the query compared to the words used to represent the objects in the video database. For example, the user may query frames involving a 'car' but the database may include only the object 'sedan'. Although 'sedan' is also a 'car', the system cannot return any frames since the object 'car' does not exactly match with the object 'sedan'.

When there is no exact match between the object in the query and the objects in the video database, it may be desirable to return approximate results. A conceptual ontology can be used in order to return approximate results. An ontology is a kind of knowledge base that involves concepts and their definitions to be used for the semantics of the application domain [34,35]. A conceptual ontology can be used to

measure the similarity between two words representing the objects. For example, 'car' and 'sedan' are semantically similar words.

Although there are different types of ontologies, we have chosen WordNet [40] ontology, since it is the most general ontology used for semantic similarity of nouns and verbs. When user queries are processed, the most similar concepts are returned to the user by evaluating semantic similarities between objects in the video database and the object in the query using the WordNet ontology. Thus our system is able to return not only the exact matches but also approximate matches by finding semantically similar objects with the help of WordNet.

5.1. WordNet ontology

WordNet which is developed at the Princeton University is a free semantic English dictionary that represents words and concepts as a network. It organizes the semantic relations between words. The minimum set of related concepts is a 'synonym set' or 'synset'. This set contains the definitions of the word sense, an example sentence and all the word forms that can refer to the same concept. For instance, the concept 'person' has a synset of {person, individual, someone, somebody, mortal, human, soul}. All these words can represent the concept 'person'.

The WordNet has an is-a hierarchy model that can be viewed as a tree having one root. A parent node is a more general term than its children. For example, 'car' is parent of 'sedan' in the WordNet hierarchy. Although most of the nodes have a single parent, some of the nodes in WordNet can have more than one parent. For this reason, WordNet is not exactly a tree [9]. In WordNet, there are nearly 75.000 concepts defined in tree-like structures where nodes are linked by relations.

We use only noun and verb hierarchies in WordNet to measure similarity between objects and similarity between activities, respectively. In addition to noun and verb concepts, hierarchies for adjectives and adverbs are also included in WordNet. We have used version 2.0 of WordNet all throughout this work.

5.2. Measuring semantic similarity between words

The aim is to use the knowledge of domain-independent semantic concepts to get better and closer results for the query. The main issue in semantic similarity is getting more accurate results between two words: the annotation word for the stored video object and the word used in the query [3]. Here semantic similarity is measured between 'words'. That is no word sense disambiguation (WSD) method is used to find the sense of the query words and video object annotation words.

The user does not enter any information about senses of the words during the annotation of videos. Therefore, all senses of both the query word and the video object should be evaluated for semantic similarity. There have been many methods for evaluating the conceptual similarity, which can be divided into three groups:

- *Distance-Based Similarity*: The methods in this group depend on counting the edges in a tree or graph based ontology [6]. Finding the shortest path is important, but when the edges are not weighted, like in WordNet, other metrics, such as the density of the graph, link type and the relation among the siblings, should also be considered.
- *Information Based Similarity*: These methods use corpus in addition to the ontology in order to get statistical values [34]. Information content is a kind of measure showing the relatedness of a concept to the domain. If the information content is high, it means the concept is more specific to the domain. For example, *school bag* has higher information content while *bag* has lower information content. Implementing these methods is more difficult than evaluating path lengths.
- *Gloss Based Similarity*: Gloss is the definition of a concept. Gloss based similarity methods depend on WordNet to find the overlapping definitions of concepts and concepts to which they are related [32]. It has the advantage that similarity between different part of speech concepts can be compared. However, gloss definitions are too short to be compared with other glosses.

In processing a query, ontology processing should constitute a small amount of the workload. Therefore, the aim is to select the fastest and relatively the most accurate method. After certain experiments with different

conceptual similarity techniques, we selected a version of Wu and Palmer's method [41] to measure the similarity between query objects and objects in the video database. Wu and Palmer's method is a distance-based similarity method. The object in the query is compared with the objects in the video database to measure their similarities. The most similar objects (the objects whose similarity values are above a certain cut-off value) are selected by our conceptual similarity method in order to be used in the semantic representation of the query.

Our conceptual similarity algorithm can find the semantic similarity degree between a query word and a stored video object word using WordNet. Since both the query word and the video object word can have many word senses, we find the similarity values for all sense pairs. The sense pair with the highest similarity value is taken as the video object's similarity to the query word. This operation is done between the query word and every video object word. Similarity values are sorted in descending order and the resulting set of objects is returned according to a cut-off value. The similarity value between a sense of the query word and a sense of a video object word is between 0 and 1, and that similarity value depends on the distance between those two senses in WordNet hierarchy. If the distance between the senses is small, the similarity value will be higher (close to 1). On the other hand, the similarity value will be low when the distance is more.

5.3. Expanding semantic representations with ontology

The information extraction module creates semantic representations of queries by using only the words appearing in queries, which are the words used for exact matches. These words are the parameters of the semantic representations, and they represent objects or activities. In order to get approximate results, the semantic representations are expanded with new words that are semantically similar to the words appearing in queries. Our conceptual similarity algorithm finds the words that are semantically similar to a query word by using the WordNet ontology.

Whenever the query includes a word representing an object, or an activity, our conceptual similarity algorithm is invoked for that word, in order to get similar words representing objects or activities in the video database. The words representing the objects and activities are stored in a result set. There may be more than one element in the result set, therefore a new semantic representation is built for each element in the set. The following example illustrates the idea:

Query: *Retrieve all frames where a car is seen.*
Semantic representation before expansion:
RetrieveObj(ObjA): frames.
ObjA(car,NULL,NULL).
Semantic representations after expansion with the ontology:
RetrieveObj(ObjA): frames.
RetrieveObj(ObjB): frames.
RetrieveObj(ObjC): frames.
ObjA(car, NULL, NULL).
ObjB(jeep, NULL, NULL).
ObjC(sedan, NULL, NULL).

In this example, we assume that our similarity algorithm finds the similarity set {*car*, *jeep*, *sedan*} for 'car'. Each word in a similarity set represents an object available in the video database, and it is semantically similar to the query word according to the WordNet ontology. The objects *car*, *jeep* and *sedan* are the objects in the video database. *Car* is retrieved since it is an exact match. *Jeep* and *sedan* are retrieved by the similarity algorithm as the most similar objects to *car*. For each element in the result set, a new semantic representation is formed. Since the similarity set is an ordered set, the order of the semantic representations is the same as the order in this similarity set. This means that *jeep* is semantically closer to *car* than *sedan* according to our similarity algorithm.

Semantic representations are expanded for not only objects but also activities. A similarity set for a verb that represents an activity is also created by the similarity algorithm, and the set is used in the expansion of the semantic representation.

Events include an activity and one or more objects. When the query includes an event with more than one object, the number of formed semantic representations increases. An ontology search is performed for all objects and the activity. According to the established semantic similarities, a result set is obtained for the activity and the object(s). The elements in the result sets are ranked depending on their semantic similarity values. Then these elements are combined to form tuples to obtain only one result set. The similarity values of the elements are multiplied so that their product represents the semantic similarity value of the tuple. The tuples are ranked depending on their similarity values in the final result set. For each tuple in the last result set, a semantic representation for the event is formed in a similar way done for the objects. As an example, consider the following query:

Query: *Retrieve all frames where John is driving a car.*
Semantic representation before expansion:
RetrieveEvnt(EvntA): frames.
EvntA(ActA, ObjA, ObjB).
ActA(drive).
ObjA(John, NULL, NULL).
ObjB(car, NULL, NULL).

Let us abstract the representation as *RetrieveEvnt(drive, John, car): frames*. Some of the similarities that are found for the words in the query by our conceptual similarity method are given as the following similarity sets:

- Similarity set of *John* is {*John*}.
- Similarity set of *car* is {*car, jeep, sedan*}.
- Similarity set of *drive* is {*drive, operate*}.

From these sets, we can find six different semantic representations. These semantic representations are ordered with respect to their computed similarity values. The similarity value of a semantic representation depends on the similarity values of the words used in the semantic representation. Thus, we get the following six abstract semantic representations by using our conceptual similarity algorithm and the WordNet ontology.

Semantic representations (abstracts) after expansion with ontology:
RetrieveEvnt(drive, John, car): frames.
RetrieveEvnt(drive, John, jeep): frames.
RetrieveEvnt(drive, John, sedan): frames.
RetrieveEvnt(operate, John, car): frames.
RetrieveEvnt(operate, John, jeep): frames.
RetrieveEvnt(operate, John, sedan): frames.

Semantic representations are constructed in order to map the query to the query processing module of the video database. The video database system has necessary functions to execute each query. These functions require the same parameters as the ones in the semantic representations. Since there is a certain semantic representation for each query type, the representations can be mapped to the functions directly. Thus the video database and the natural language query interface are independent systems that communicate through the semantic representations.

Whenever a query is posed to the natural language interface, it returns not only the semantic representation of the query but also the type of the query to the video database. Since the parameters of the functions and the representations are the same, the system calls the appropriate functions to execute the query.

6. Evaluation

In order to evaluate the effectiveness of our ontology-based querying method, we created a test domain. This test domain consists of videos of a TV serial and videos obtained from a street camera. We especially

put videos from different domains in order to show that our method is domain-independent. Video objects are annotated with words by our annotation tool [22]. The total length of videos in the test domain was 280 min, and there were 710 distinct video objects that were annotated with nouns.

We prepared a set of queries and made a list of all words that were used in these queries. The total number of query words was 300. Some of these query words were the same as the words used to describe the objects in the database. In other words, they would correspond to exact matches. The 83% of the query words did not appear in the database. A human expert decided the correct frames for each query. The human expert marked not only the exact matches but also the video frames that could match semantically with the query word. For instance, for the query word 'vehicle', the human expert marked all video frames containing objects described by words such as 'car', 'bus', 'sedan', 'plane', 'ship' as correct answers. The human expert also considered the intended meanings of the query words in marking the correct answers. For instance, if the intended meaning of the query word 'bat' is baseball bat, all video frames containing a baseball bat are marked as correct answers but not the video frames containing the animal 'bat'.

Then we executed the prepared queries to see which video frames would be retrieved by the system. For each query, the returned answer set was compared with the correct answers prepared by the human expert, in order to evaluate the success of our querying mechanism. The accuracy was measured by using the well-known metric F-measure which was evaluated by the following formulas:

$$\text{precision} = \frac{\text{number of correct answers in the answer set}}{\text{number of all answers in the answer set}}$$

$$\text{recall} = \frac{\text{number of correct answers in the answer set}}{\text{number of all possible correct answers}}$$

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

Each query was expanded by using the words that are similar to the words appearing in the query. The amount of the expansion depends on the cut-off value used by our conceptual similarity algorithm. We got different expansions for different cut-off values and we got different answers for these different expansions. The test results for the test domain are given in Fig. 5. For the test domain, we got the best accuracy result (F-measure) when the cut-off value was 0.80. Our accuracy results are as follows when the cut-off value is 0.80:

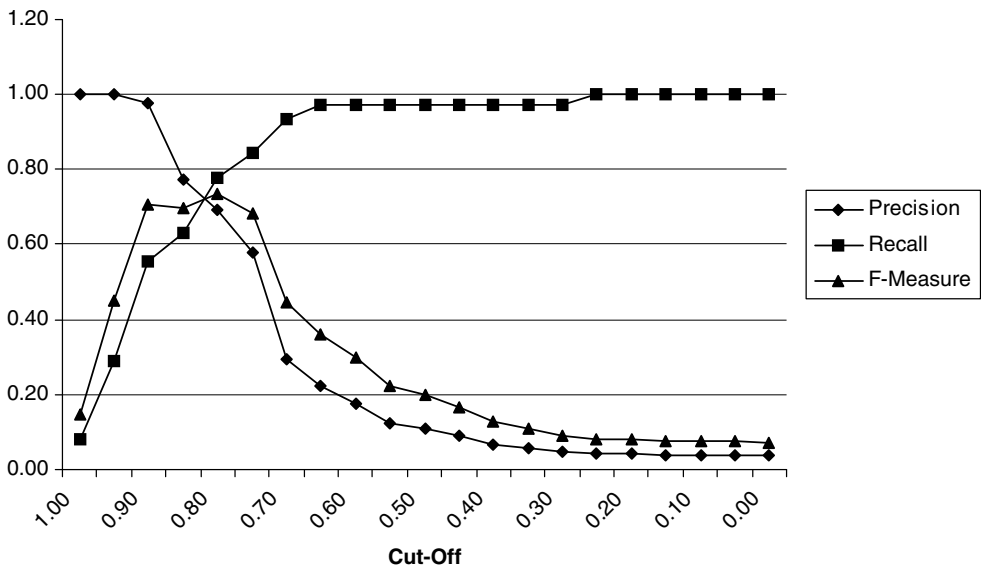

Fig. 5. Test results for the test domain (non-sensed).

F-measure 0.73
Precision 0.69
Recall 0.78

These results indicate that 69% of the results in the answer set are correct, and 78% of all possible correct answers appear in the answer set when the cut-off value is 0.80. These are the results with respect to the best *F*-measure value (0.73). Of course, the precision can increase when we increase the cut-off value, but the recall drops in this case. If the cut-off value comes close to 1.0, the precision becomes 1.0 too while the recall drops to its minimum (8%). The recall reaches to its maximum when the cut-off value comes close to 0, but the precision drops to its minimum (4%).

A noun can have multiple senses (meanings). Our similarity algorithm assumes that the query word is related to a video object word, if a sense of the query word is similar to a sense of the video object word. The query word can be related to an unintended sense of the video object word. In this case, they will be treated as similar words, and the video frame containing that video object word will be selected into the answer set. For example, the word 'mammal' is related with the animal sense of the word 'bat', and it is unrelated with the baseball bat sense of the word 'bat'. When the video frames containing a mammal are searched, the video frame containing a baseball bat can also appear as an incorrect result in the answer set, just because the video frame containing a baseball bat is annotated with the word 'bat' only.

When a video object is annotated with a word, the sense of the word is actually known by the annotator. If the annotator records the intended sense of the word together with the word, the precision and the recall will increase. We wanted to know how much improvement we would get in our results if the video objects were annotated with words together with their intended senses. We repeated the experiment with a test domain in which words were recorded together with their intended senses during annotation. When we executed test queries with this re-annotated test domain, the results were dramatically improved. We got the following results in this experiment when the cut-off value was 0.80:

F-measure 0.91
Precision 0.88
Recall 0.95

This experiment indicates that an extra effort in the annotation dramatically improves the result. The results of this experiment are given in Fig. 6.

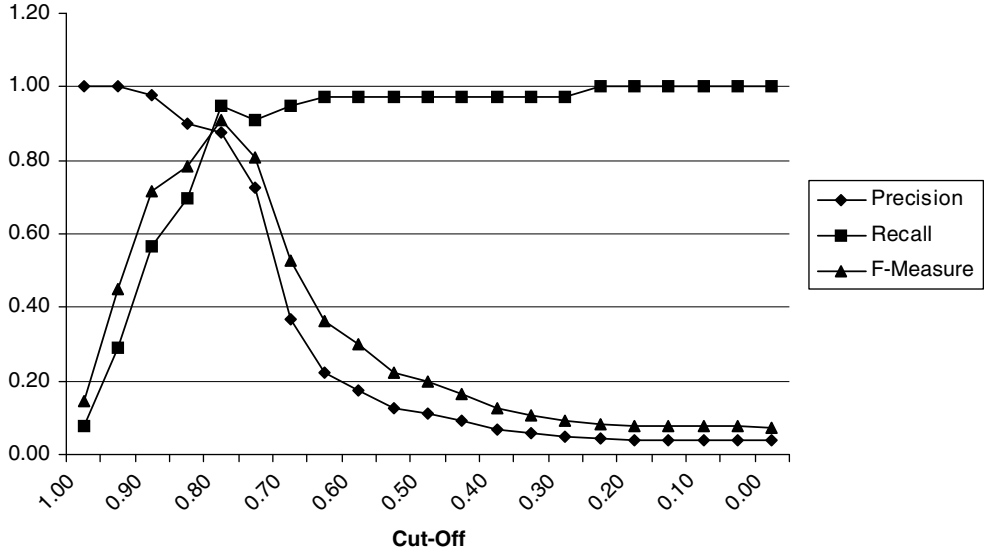

Fig. 6. Test results for the test domain (sensed).

In the literature, there are not too many systems that use a natural language interface with their accuracy results reported in their papers. The system described in [43] reports that their precision value for their tests is 0.372, and their recall value is 0.626. Of course, since their test domain and our test domain are not same, it may not be fair to compare the quantitative results of two systems.

7. Conclusion

The system described in this paper uses a natural language querying interface to retrieve information from a video database which supports content-based spatio-temporal querying. In order to implement the natural language interface, a light-parsing algorithm is used to parse queries and an information extraction algorithm is used to find the semantic representations of the queries. The core part of the extraction step is the detection of objects, events, activities and spatio-temporal relations. The semantic representation is constructed as the result of parsing the sentence with the link rules in a knowledge base. The semantic representation is used to map the query to the functions of the query processing module of the video database system. A conceptual ontology search is implemented as part of the natural language interface. Using the ontological structure, WordNet, the system retrieves the most similar objects or activities to the words given in the query. An edge-based method is combined with corpus-based techniques in order to get higher accuracy from the system. The semantic representations enriched with the ontology are sent to the video database system to call the appropriate query function.

In the current semantic representation of objects, an object can have only limited number of simple attributes. As a future extension, we are planning to add more complex attributes to describe the objects. Adding more complex attributes means that we have to deal with more complex noun phrases in the queries. The information extraction module will then be more complex for objects; however the querying capability of the system will have been increased. When the video database is expanded to handle compound and conjunctive query types, the extraction rules will be expanded to handle more complex queries.

Ontology related experiments show us that if the user annotates the video with not only plain words but also their senses in the WordNet, the accuracy rate of the natural language processing would increase. In a future study, we plan to expand the annotations in video database with senses attached to the words describing the entities. This extension can increase the annotation cost of videos, but it will increase the accuracy of the results.

Acknowledgements

This work is partially supported by The Scientific and Technical Council of Turkey Grant "TUBITAK EEEAG-107E234".

References

[1] B. Acharya, A.K. Majumdar, J. Mukherjee, Video model for dynamic objects, Information Sciences 176 (17) (2006) 2567–2602.
[2] S. Adali, K.S. Candan, S. Chen, K. Erol, V.S. Subrahmanian, The advanced video information system: data structures and query processing, Multimedia Systems 4 (1996) 172–186.
[3] T. Andreasen, H. Bulskov, R. Knappe, On ontology-based querying, in: H. Stuckenschmidt (Eds.), Proceedings of the 18th International Joint Conference on Artificial Intelligence, Ontologies and Distributed Systems (IJCAI 2003), Acapulco, Mexico, 2003, pp. 53–59.
[4] I. Androutsopoulos, G. Ritchie, P. Thanisch, MASQUE/SQL – An efficient and portable natural language query interface for relational databases, in: Proceedings of the Sixth International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, Edinburgh, 1993, pp. 327–330.
[5] I. Androutsopoulos, G. Ritchie, P. Thanisch, Natural language interfaces to databases – an introduction, Journal of Natural Language Engineering 1 (1) (1995) 29–81.
[6] A. Budanitsky, G. Hirst, Semantic distance in WordNet: an experiment, application-oriented evaluation of five measures, in: Proceedings of NAACL 2001 – WordNet and Other Lexical Resources Workshop, Pittsburgh, 2001, pp. 29–34.
[7] S.F. Chang, W. Chen, H.J. Meng, H. Sundaram, D. Zhong, VideoQ: an automated content based video search system using visual cue, in: Proceedings of ACM International Conference on Multimedia'97, Seattle, WA, November 9–13, 1997, pp. 313–324.
[8] C. Decleir, M.S. Hacid, J. Kouloumdjian, Modeling and querying video databases, in: Proceedings of Conference EUROMICRO, Multimedia and Communication Track, Vastras, Sweden, 1998, pp. 492–498.

[9] A. Devitt, C. Vogel, The topology of WordNet: some metrics, in: P. Sojka, K. Pala, P. Smrz, C. Fellbaum, P. Vossen (Eds.), Proceedings of GWC 2004, Masaryk University, Brno, 2003, pp. 106–111.
[10] M.E. Donderler, E. Şaykol, U. Arslan, O. Ulusoy, U. Gudukbay, BilVideo: design and implementations of a video database management system, Multimedia Tools and Applications 27 (1) (2005) 79–104.
[11] F. Durupinar, U. Kahramankaptan, I. Cicekli, Intelligent indexing, querying and reconstruction of crime scene photographs, in: Proceedings of TAINN2004, Izmir, Turkey, 2004, pp. 297–306.
[12] G. Erozel, Natural language interface on a video data Model, MS Thesis, Department of Computer Engineering, METU, Ankara, 2005.
[13] G. Erozel, N.K. Cicekli, I. Cicekli, Natural language interface on a video data model, in: Proceedings of IASTED International Conference on Databases and Applications (DBA2005), Innsbruck, Austria, 2005, pp. 198–203.
[14] T.R. Gayatri, S. Raman, Natural language interface to video database, Natural Language Engineering 7 (1) (2001) 1–27.
[15] M.C. Hacid, C. Decleir, J. Kouloumdjian, A database approach for modeling and querying video data, IEEE Transactions on Knowledge and Data Engineering 12 (5) (2000) 729–750.
[16] R. Hjelsvold, R. Midtstraum, Modeling and querying video data, in: Proceedings of the 20th VLDB Conference, Santiago, Chile, 1994, pp. 686–694.
[17] R. Hjelsvold, S. Lagorgen, R. Midtstraum, O. Sandsta, Integrated video archive tools, in: Proceedings of ACM International Conference on Multimedia'95, San Francisco, CA, November 5–9, 1995, pp. 283–293.
[18] Informedia, Carnegie Mellon University, Available from: <http://www.informedia.cs.cmu.edu>.
[19] B. Katz, J. Lin, C. Stauffer, E. Grimson, Answering questions about moving objects in surveillance videos, in: Proceedings of AAAI Symposium on New Directions in Question Answering, Palo Alto, California, 2003.
[20] B. Katz, S. Felshin, D. Yuret, A. Ibrahim, J. Lin, G. Marton, A.J. McFarland, B. Temelkuran, Omnibase: Uniform access to heterogeneous data for question answering, in: Proceedings of the 7th International Workshop on Applications of Natural Language to Information Systems (NLDB 2002), 2002.
[21] R. Knappe, H. Bulskov, T. Andreasen, Perspectives on ontology-based querying, in: H. Stuckenschmidt (Eds.), Proceedings of the 18th International Joint Conference on Artificial Intelligence, Ontologies and Distributed Systems (IJCAI 2003), Acapulco, Mexico, 2003.
[22] M. Koprulu, N.K. Cicekli, A. Yazici, Spatio-temporal querying in video databases, Information Sciences 160 (2004) 131–152.
[23] O. Küçüktunç, U. Güdükbay, O. Ulusoy, A natural language-based interface for querying a video database, IEEE Multimedia – Multimedia at Work 14 (1) (2007) 83–89.
[24] T.C.T. Kuo, A.L.P. Chen, Content-based query processing for video databases, IEEE Transactions on Multimedia 2 (1) (2000) 1–13.
[25] H. Lee, A.F. Smeaton, J. Furner, User interface issues for browsing digital video, in: Proceedings of the 21st BCS IRSG Colloquium on IR, Glasgow, 1999.
[26] J. Li, M. Özsu, D. Szafron, V. Oria, Multimedia Extensions to Database Query Languages, Technical Report TR-97-01, Department of Computing Science, The University of Alberta, Alberta, Canada, 1997.
[27] W. Li, S. Gauch, J. Gauch, K. Pua, VISION: a digital video library, in: Proceedings of ACM International Conference on Digital Libraries (DL'96), Bethesda, MD, 1996, pp. 19–27.
[28] LinkParser, Available from <http://www.link.cs.cmu.edu/link>.
[29] V. Lum, D.A. Keim, K. Changkim, Intelligent natural language processing for media data query, in: Proceedings of International Golden West Conference on Intelligent Systems, Reno, Nevada, 1992.
[30] E. Oomoto, K. Tanaka, OVID: design and implementation of a video object database system, IEEE Transaction on Knowledge and Data Engineering 5 (4) (1993) 629–643.
[31] K. Pastra, H. Saggion, Y. Wilkis, Extracting relational facts for indexing and retrieval of crime-scene photographs, Knowledge-Based Systems 16 (5–6) (2002) 313–320.
[32] T. Pedersen, S. Banerjee, S. Pathwardan, Maximizing semantic relatedness to perform word sense disambiguation, University of Minnesota Supercomputing Institute, Research Report UMSI 2005/25, March, 2005.
[33] A.L. Ramshaw, M. Marcus, Text chunking using transformation-based learning, in: Proceedings of the ACL Third Workshop on Very Large Corpora, 1995, pp. 82–94.
[34] P. Resnik, Semantic similarity in a taxonomy: an information-based measure and its application to problems of ambiguity in natural language, Journal Artificial Intelligence Research 11 (1999) 95–130.
[35] M.A. Rodriguez, M.J. Egenhofer, Determining semantic similarity among entity classes from different ontologies, IEEE Transactions on Knowledge and Data Engineering 15 (2003) 442–465.
[36] D. Sleator, D. Temperley, Parsing English with a link grammar, in: Proceedings of the Third International Workshop on Parsing Technologies, 1993.
[37] D. Swanberg, C.F. Shu, R. Jain, Knowledge guided parsing in video databases, in: W. Niblack (Ed.), Proceedings of SPIE, Storage and Retrieval for Image and Video Databases, vol. 1908, San Jose, California, 1993, pp. 13–24.
[38] T. Vanrullen, P. Blache, An evaluation of different shallow parsing techniques, in: Proceedings of LREC-2002, 2002.
[39] W.A. Woods, R.M. Kaplan, B.N. Webber, The lunar sciences natural language information system: Final report, BBN Report 2378, Bolt Beranek and Newman Inc., Cambridge, Massachussetts, 1972.
[40] WordNet 2.1, Available from: <http://wordnet.princeton.edu/online/>, 2005.
[41] Z. Wu, M. Palmer, Verb Semantics and lexical selection, in: Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, Las Cruces, New Mexico, 1994, pp. 133–138.

[42] H. Yang, C. Lekha, Y. Zhao, S. Neo, T. Chua, VideoQA: question answering in news video, in: Proceedings of the 11th ACM MM, Berkeley, USA, 2003, pp. 632–641.
[43] D. Zhang, J.F. Nunamaker, A natural language approach to content-based video indexing and retrieval for interactive e-learning, IEEE Transaction Multimedia 6 (3) (2004) 450–458.
[44] H.J. Zhang, C.Y. Low, S.W. Smoliar, J.H. Wu, Video parsing, retrieval and browsing: an integrated and content-based solution, in: Proceedings of ACM International Conference on Multimedia'95, San Francisco, CA, November 7–9, 1995, pp. 503–512.