Natural Language Interface on a Video Data Model

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

[a] *Department of Computer Engineering, Middle East Technical University, Ankara, 06531, Turkey,*
e131262@ceng.metu.edu.tr, nihan@ceng.metu.edu.tr

[b] *Department of Computer Engineering, Bilkent University, Ankara, Turkey, ilyas@cs.bilkent.edu.tr*

Abstract

Depending on a content-based spatio-temporal video data model, a natural language interface is implemented to query the video data. The queries, which are given as English sentences, are parsed using Link Parser, and the semantic representations of given queries are extracted from their syntactic structures using information extraction techniques. At the last step, the extracted semantic representations are used to call the related parts of the underlying spatio-temporal video data model to get the results of the queries.
Keywords: Natural Language Querying, Spatio-Temporal Video Databases, Link Parser, Information Extraction.

1. Introduction

Multi-media data models and databases are subjects of the recent research interests. Implementing video data models and methods to retrieve data are some of the main concerns on video databases [5, 6, 8, 11]. Unlike relational databases, spatio-temporal properties and rich set of semantic structures make querying and indexing video data more complex. Therefore, all known formal query languages become ineffective for video data retrieval. New effective techniques, which one of them is natural language (NL) interface, are implemented for querying this type of data. Depending on video data structures, text-based and/or graphical user interfaces are used in querying [9, 10, 12]. In text-based systems, annotation extraction, hierarchical design and natural language processing are used contrasting to graphical methods as pattern matching and trajectory drawing.

There are various methods for retrieving the video data. These methods include ontology querying, annotation-based structures, content-based structures, rule-based querying and some specific SQL-like languages[2]. Each method has its own advantages and disadvantages. In this study, natural language interface for querying is used in order to provide a flexible system where the user can use his/her own sentences. The user does not have to learn an artificial query language, which is a great advantage of natural language processing (NLP) [1]. NLP sometimes is the most flexible way of expressing queries over complex data models. Elliptical and anaphoric statements can be seen as the proof of this flexibility. On the other hand, there are still some disadvantages that users are limited by the domain and by the capabilities of parsers; so 100% accuracy cannot be achieved. But in recent studies, we see that NLP techniques are improved and it is possible to obtain approximately 90% accuracy. As related work, there are other projects that use NLP techniques in querying video data. They use syntactic parsers to convert the media descriptions or annotations to be stored and build semantic ontology trees from the parsed query [10, 15].

The aim of this paper is to describe the implementation of the NL interface over the specified video data. In the NLP part, queries are parsed, and their semantic representations are extracted from their syntactic structures. In order not to concern with the entire parse tree, a light parsing algorithm is chosen for implementing the interface.

The rest of the paper is organized as follows. The video data model and its previous query interface are explained briefly in Section 2. In Section 3, we discuss our solution to implement an NLP interface in detail. Finally Section 4 is the conclusion and future work.

2. Video Data Model

The video data model on which the natural language interface is implemented is a content based spatio-temporal video data model [11]. The basic elements of the data model are *objects*, *activities* and *events*. The video clip is divided into a time-based partition called frames. Frames have the time interval equal to minutes. Objects are the 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 the verbal clauses like playing football, singing etc. Events are detailed activities that are constructed from an activity name and an object with some role. For instance *John plays football*, and *the cat is catching a mouse* are events.

In this video data model, spatio-temporal queries are the main concern. Spatial properties are considered in a two-dimensional space. 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. Other types of queries are trajectory and occurrence queries. 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. Occurrence queries are basic kinds of queries asking to retrieve the frames for a given object, event or an activity. In addition, the occurrence queries include queries to retrieve objects, events and activities for a given time interval. The query types that the video data model supports are presented in the first column and their examples are in the third column of Table 1.

In the previous implementation of our video database, a graphical user interface was used to query the system. Pull-down menus and buttons were used to select objects, events, activities and spatial relations to express a query. When a spatial relation is queried, related objects, spatial relation and also a threshold value were chosen from the drop down lists, and the type of the query must be selected using buttons. But this interface has been very restricted for the user and also for the project itself. Since it would be time consuming to extend the model for future applications, we have decided to use a natural language interface for querying.

3. Query Processing

Instead of using the restricted graphical user interface for queries, a natural language interface is decided to be used for the flexibility. The idea is to map the English sentence 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 data model to process the query and show the results. The main structure of the system is given in Figure 1. In the rest of this section, the querying system using natural language is explained in detail.

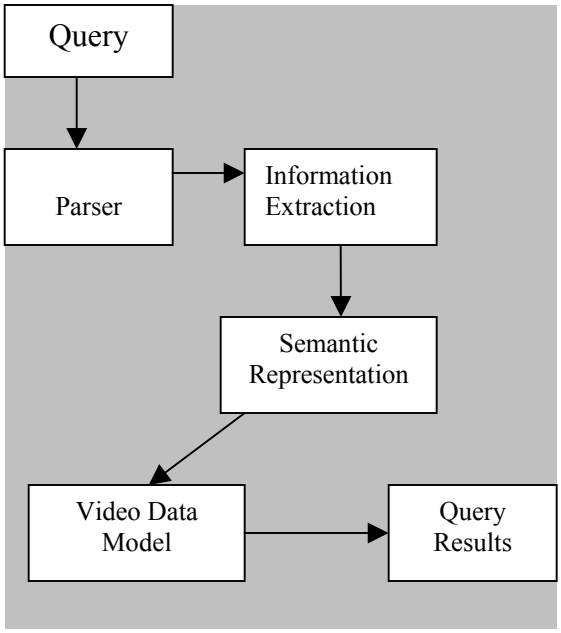

Figure 1. System Design

3. 1. Semantic Representations

Since only certain kinds of queries are used in our video data model, it is sufficient to find the type of a given query and its parameters to extract a semantic representation of the query. The structure of the semantic representations is made similar to the underlying data model structures. Therefore, in order to obtain a semantic representation of a query given as a natural language sentence, 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.

Every query should include at least an object, or an event, or an activity as in the structure of the video data model. Object and activity are atomic particles that sometimes form an event. Objects also can 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 parameters. So, 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 following objects are the actors of this activity. When the full semantic representation of a query is tried to be constructed; the activity and objects are extracted in the next step after event extraction.

There are also other kinds of semantic representations for spatial and temporal properties in the query. Some of them are atomic structures using 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 these rectangles must be converted to two dimensional coordinates in order to map into the functions of our data model. Thus, regional semantic representation is:

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

where 'Region' is the predicate name that will be used in the query semantic representation. 'x1' and 'y1' are the upper left corner; 'x2' and 'y2' are the right-down corner of the regional rectangle.

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 representing these spatial relations, 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)*
- *UPPER-LEFT (object1, object2, threshold)*
- *LEFT (object1, object2, threshold)*
- *UPPER-RIGHT (object1, object2, threshold)*

In these predicates, 'threshold' value is used to specify the fuzziness in the spatial relations.

Each query in Table 1 has a different semantic representation, and they have a different set of parameters in their semantic representations. So, the extractions depend on the type of the query. The semantic representations of the parameters are extracted, and they are combined to get the semantic representation of the query.

3.1. Mapping Queries to Semantic Representations

A light parsing algorithm has been chosen to parse the queries because only specific kinds of word groups (like objects, activities, start of the interval etc.) are needed to obtain the semantic representations. Since there is no need to find the whole detailed parse tree of a query, a light parsing algorithm such as shallow parser [16, 19], chunk parser [4, 18] and link parser [13, 20] is enough for our purposes. We have chosen to use a link parser to parse given queries in our implementation.

Described in [13, 20], 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 does not cross - . In the following example, Ds is a link that connects the singular determiner with its noun.

```
|----Ds---|
a        cat
```

As seen in Figure 2, some of the words in a parsed query are associated with their part of speech tags such as noun (.n) and verb (.v). Then the word groups in the sentence are connected with linkages and each linkage has also a type.

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 [7] which has been adopted from information extraction methodology used in SOCIS [17]. In [7], crime photos are indexed by the relations and scene descriptions by using the information extraction in an application domain. In our system word groups in a parsed query are mapped to the specific parts of the semantic representation of that query, such as *objects*, *activities, intervals, regions* and *spatial relations*. 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.

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 find an activity:

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 Representations of Examples
Elementary Object Queries	RetrieveObj (objA) : *frame_list*	Retrieve all frames in which Bush is seen.	-RetrieveObj (Obj_A): *frames.* -Obj_A (Bush, 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 minutes in the clip.	-RetrieveIntObj (Int_A): *objects.* -Int_A(x-5, x). [x: Temporal length of video]
Activity Type Occurrence Queries	RetrieveIntAct (intervalA) : *activity_list*	Retrieve activities performed in the first 20 minutes.	-RetrieveIntAct (Int_A): *activities.* -Int_A (0, 20).
Event Occurrence Queries	RetrieveIntEvt (intervalA) : *events_list*	Find all events performed in the last 10 minutes	-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.* -Evnt_A (Act_A, Obj_A, NULL). -Act_A (run). -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 minutes.	-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]

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

For each query, first the query type is extracted from the parsed query. Then, the parts of the semantic representation are extracted. For example, if the query is a trajectory query, an object, a start region, and an end region should be found. So, the rule for finding an object path is traced from the linkage. Then, to find the other elements, other linkages are traced. In Figure 2, Op linkage helps us to determine the type of the query as *Elementary Object Query*, then we need to find the object involved in this query type. The link orders and G linkages help us to determine this object as *Elton John*.

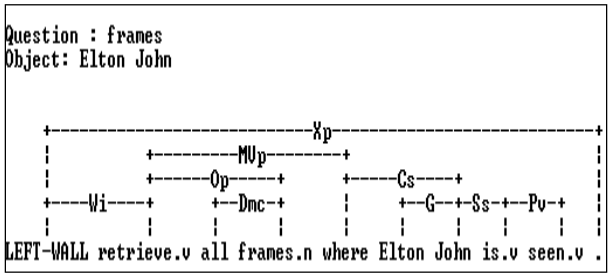

Figure 2. Example of a query parsing with link grammar and semantic representation.

Certain parts in the parsed query may not be directly mapped into a part of the semantic representation. For example, a numerical value can be entered either as a number or as a word phrase in a given query (such as *1* versus *one*), but in 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 or sides relative to the screen like *left, center, upper left* etc. To map this data with the video data model, these areas should be converted into two dimensional coordinates. Thus, the regions can be represented as rectangles. So, the screen is thought to be divided into 5 regions as upper-right, upper-left, down-left, down-right and center. Depending on the area in query, it is matched with these regions. For example, if in the query, 'right' area is asked as the region, then the coordinates of upper right + down-right are evaluated.

A similar problem also occurs in interval queries. When the user enters as *last 10 minutes*, the beginning time must be evaluated to map with the video data. Therefore, the extraction algorithm is also responsible for these conversions.

As a result, depending on the rules for the query word (what is asked), the parts of the query such as object, activity, event, interval, relation, threshold value and region are extracted from the parsed query. And gathering these elements, a semantic representation for each query is obtained.

4. Conclusion and Future Work

The system described in this paper uses a natural language interface to retrieve information from the video database. For this purpose, a light parsing algorithm is used to parse queries and an extraction algorithm is used to find the semantic representations of the queries. Detection of objects, events, activities and relations is the core part of the extraction step. When the sentence is parsed with decided link rules, the semantic relation is constructed depending on the type of the query. This process will be used for mapping the semantic representation over the functions of the video data model.

As a future extension, we are planning to add more complex attributes to describe the objects. In the current semantic representation of objects, an object can have only two attributes. Adding more complex attributes means we have to deal with more complex noun phrases. The information extraction module will then be more complex for objects, however the querying ability of the user will have been increased.

In order to get better accuracy in the results of the queries, a conceptual ontology will be added to the algorithm by using WordNet. The ontological search tree will be used for objects, activities (indirectly events) and spatial relations using a similarity search algorithm on ontology tree. An approach for this algorithm should be finding highest similarity degree by using the generalization and specification factors like in [1].

References

[1] Andreasen T, Bulskov H, Knappe R, "On Ontology-Based Querying", *pp. 53-59 in Heiner Stuckenschmidt (Eds.): 18th International Joint Conference on Artificial Intelligence, Ontologies and Distributed Systems, IJCAI 2003, Acapulco, Mexico, August 9 to 15, 2003.*

[2] Androutsopoulos I, Ritchie G, Thanisch P, "MASQUE/SQL – An Efficient and Portable Natural Language Query Interface for Relational Databases", *Proceedings of the Sixth International Conference on Industrial & Engineering Applications of Artificial Intelligence and Expert Systems, Edinburgh*, 1993.

[3] Androutsopoulos I, Ritchie G, Thanisch P, "Natural Language Interfaces to Databases", *Journal of Natural Language Engineering, Cambridge University Press*, 1994.

[4] Brooks P, "SCP: A Simple Chunk Parser", *Artificial Intelligence Center, The University of Georgia Athens, Georgia*, 2003.

[5] Decleir C, Hacid M.S, Kouloumdjian J, "Modelling and Querying Video Databases", *Conference EUROMICRO, Multimedia and Communication Track, Vastras, Sweden, pp 492-498*, 1998.

[6] Donderler M.E, Şaykol E, Arslan U, Ulusoy O, "BilVideo: Design and Implementations of a Video Database Management System", *Kluwer Academic Publishers*, 2003.

[7] Durupinar F, Kahramankaptan K, Cicekli I, "Intelligent Indexing, Querying and Reconstruction of Crime Scene Photographs", *in Proc. Of TAINN2004*, 2004.

[8] Hjelsvold R, Midtstraum R, "Modeling and Querying Video Data", *20th VLDB Conference Santiago, Chile*, 1994.

[9] Informedia *(Carnegie Mellon University)*, http://www.informedia.cs.cmu.edu/html/description.html

[10] Katz B, Lin J, Stauffer C, Grimson E, "Answering Questions about Moving Objects in Surveillance Videos", *American Association for Artificial Intelligence*, 2002.

[11] Koprulu M, Cicekli N.K, Yazici A, "Spatio-temporal Querying in Video Databases", *Information Sciences 160, 2004, pp.131-152*.

[12] Lee H, "User-Interface for Digital Video Systems", *Technical Report*, 1998.

[13] Link Parser, http://www.link.cs.cmu.edu.tr/link/

[14] Loper E, Bird S, "NLTK Tutorial: Chunking", *Creative Commons*, 2004.

[15] Lum V, Keim D.A, Changkim K, "Intelligent Natural Language Processing for Media Data Query", *Proc. Int. Golden West Conf. on Intelligent Systems, Reno, NEV.*, 1992

[16] Munoz M, Punyakanok V, Roth D, Zimak D, "A Learning Approach to Shallow Parsing", 2000.

[17] Pastra K, Saggion H, Wilkis Y, "Extracting Relational Facts for Indexing and Retrieval of Crime-Scene Photographs", *Knowledge-Based Systems, vol. 16 (5-6), pp.313-320, Elsevier Science*, 2002.

[18] Ramshaw A.L, Marcus M, "Text Chunking Using Transformation-based Learning", *In Proceedings of the ACL Third Workshop on Very Large Corpora*, *pp. 82-94*, 1995.

[19] Rullen T, Blache P, "An evaluation of Different Shallow Parsing Techniques", *in proceedings of LREC-2002.*

[20] Sleator D, Temperley D, "Parsing English with a Link Grammar", *Third International Workshop on Parsing Technologies*, 1993.