NATURAL LANGUAGE INTERFACE ON A VIDEO DATA MODEL

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

GÜZEN ERÖZEL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

JULY 2005

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. Canan Özgen
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Ayşe Kiper
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Nihan K. Çiçekli
Supervisor

Examining Committee Members

Assoc. Prof. Dr. İ. Hakkı Toroslu (METU, CENG)

Assoc. Prof. Dr. Nihan Kesim Çiçekli (METU, CENG)

Assoc. Prof. Dr. İlyas Çiçekli (Bilkent University)

Assoc. Prof. Dr. Ahmet Coşar (METU, CENG)

Dr. Meltem Turhan Yöndem (METU, CENG)

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name:

Signature :

ABSTRACT

NATURAL LANGUAGE INTERFACE ON A VIDEO DATA MODEL

Erözel, Güzen

M.Sc., Department of Computer Engineering

Supervisor: Assoc. Prof. Dr. Nihan Kesim Çiçekli

July 2005, 107 pages

The video databases and retrieval of data from these databases have become popular in various business areas of work with the improvements in technology. As a kind of video database, video archive systems need user-friendly interfaces to retrieve video frames. In this thesis, an NLP based user interface to a video database system is developed using 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, NL interface enables flexible querying. The queries, which are given as English sentences, are parsed using Link Parser. Not only exact matches but similar objects and activities are also returned from the database with the help of the conceptual ontology module to return all related frames to the user. This module is implemented using a distance-based method of semantic similarity search on the semantic domain-independent ontology, WordNet. The semantic representations of the given 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 spatio-temporal video data model to calculate the results of the queries.

Keywords: Natural Language Querying, Spatio-Temporal Video Databases, Link Parser, Information Extraction, Conceptual Ontology.

ÖZ

BİR VİDEO VERİ MODELİ ÜZERİNDE TANIMLANAN DOĞAL DİL ARAYÜZÜ

Erözel, Güzen

Yüksek Lisans, Bilgisayar Mühendisliği Bölümü

Tez Yöneticisi: Doç. Dr. Nihan Kesim Çiçekli

Temmuz 2005, 107 sayfa

Video veritabanları ve bu veritabanlarından bilginin elde edilmesi teknolojinin gelişmesiyle çeşitli iş alanlarında popüler olmaya başlamıştır. Bir çeşit video veritabanı olan video arşiv sistemleri video sahnelerini getirmek için kullanıcı dostu arayüzlere ihtiyaç duymaktadır. Bu tezde içerik bazlı uzaysal-süreksiz bir video veri modeli kullanan bir video veritabanı sistemine doğal dil tabanlı kullanıcı arayüzü geliştirilmiştir. Bu veri modelinde nesneler, bu nesnelerin gerçekleştirdiği aktiviteler ve nesnelerin uzaysal özelliklerini içeren anlamsal içeriğe odaklanılmıştır. Bu model ile nesneler arasındaki uzaysal-süreksiz ilişkilerin ve hareketli nesnelerin yörüngelerin sorgusu yapılabilmektedir. Bu video veritabanı sisteminde doğal dil arayüzü daha esnek sorgulama yapılabilmesini sağlamaktadır. İngilizce olarak verilen sorguların, Link Ayrıştırıcısı tarafından sözdizimsel analizi yapılır. Kavramsal ontoloji modülü sayesinde sadece kesin uyuşanlar değil, ayrıca benzer nesne ve aktivitelerin veritabanından getirilmesi sağlanarak kullanıcıya tüm ilişkili sahneler sunulabilmektedir. Bu modül, anlamsal ve alan-bağımsız bir ontoloji olan WordNet üzerinde uzaklık-bazlı bir anlamsal benzerlik araştırma metodu kullanılarak gerçekleştirilmiştir. Sorguların anlamsal gösterimleri sorguların sözdizimsel yapılarından bilgi çıkarım teknikleri kullanılarak oluşturulmuştur. Çıkarımı yapılan bu anlamsal gösterimler, sistemin temelini oluşturan uzaysal-süreksiz veri modelinin ilgili parçalarını çağırarak sorgu sonuçlarını hesaplamak için kullanılır.

Anahtar Sözcükler : Doğal Dil Sorgulama, Uzaysal-Süreksiz Veri Tabanları, Link Ayrıştırıcı, Bilgi Çıkarımı, Kavramsal Ontoloji.

To my family...

ACKNOWLEDGMENTS

I would like to present my special thanks to my supervisor Assoc. Prof. Dr. Nihan Kesim Çiçekli for her guidance, understanding and encouragement throughout the development of this thesis. Also my special thank is to Asst. Prof. Dr. İlyas Çiçekli, who guided and helped me for this thesis.

I would also like to thank to my employer, Central Bank of the Republic of Turkey, for providing me time for any MSc. study whenever I needed.

I thank all of my friends for their patience and unforgettable help. Finally, my deepest thanks are to my parents who supported and motivated me with their never ending patience, tolerance, understanding and love throughout this demanding study.

TABLE OF CONTENTS

CHAPTER

APPENDICES

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

API	: Application Programmer’s Interface
AVIS	: Advanced Video Information System
MBR	: Minimum Bounding Rectangles
MMDBMS	: Multimedia Database Management System
NL	: Natural Language
NLP	: Natural Language Processing
NP	: Noun Phrase
POS	: Part of Speech
QBIC	: Query By Image Content
QBE	: Query By Example
SQL	: Structured Query Language
VP	: Verb Phrase
WSD	: Word Sense Disambiguation

CHAPTER 1

INTRODUCTION

It has been possible to archive and query multimedia data in computers as a result of recent technological developments. The current technological developments offer convenient ways of saving and querying video files that include movies, news clips, sports events, medical scenes, security camera recordings to the 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 catalog information. The second problem is how to query the content of the videos in a detailed and easy manner.

Unlike relational databases, spatio-temporal properties and rich set of semantic structures make querying and indexing video data more complex. Due to the complexity of video data, there have been many video data models proposed for video databases [11, 13, 18, 24]. Some of the existing work use annotation based modeling. Some use physical level video segmentation approach [51], 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, 37]. 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. We can 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, SWIM and VideoQ [26]. These are in general easy to use systems but they are not flexible enough. Only specific query types can be asked by the user with these interfaces. For instance, with trajectory drawing only trajectory queries can be handled. On the other hand, textual interfaces that require the user to enter the queries with SQL-like query languages or extensions to SQL are difficult to use, since the user has to learn the syntax of the language [4, 13]. 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 and VISION [27]. The most flexible method among all these approaches is natural language.

The aim of the thesis is to build a natural language query interface over a previously developed video data model [24] which has many querying capabilities that other video data models do not have. The video data model 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 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 allows inclusion of fuzziness in spatial and spatio-temporal queries.

The main contribution of this thesis is the capability of querying the system in a natural language instead of using an artificial language. For instance, the following kinds of queries will be possible in this system:

- *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.* (in querying a security camera recording)

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 to be stored and build semantic ontology trees from the parsed query [33, 39]. But these are usually application specific and domain dependent (e.g. querying only the recordings of street cameras in SPOT [21] or querying only the parts of news broadcast in Informedia [19]). Not every system using natural language can capture high-level semantics. The video system Informedia using keyword-matching natural language interface cannot answer detailed queries nor handle structures with attributes [19]. However our system can handle compound structures and different types of questions. In this thesis we are aiming at implementing a general purpose video database querying system by adding a natural language interface to a previously developed video data model.

Another contribution of the querying facility of the system is using information extraction techniques to find the semantic representation of user queries [15]. It is preferred to extract sentences to map with the data model in studies using natural language. In SOCIS system, the crime scene photographs are described using NLP and extracted to index the photos [14, 39]. However, only spatial relations are extracted in this system. But in our system, all the query types are extracted to build their semantic representations to map with the video database.

A more important contribution of this thesis is to perform an ontological search by using the ontological structures of words. In querying, the system will not only search the given words but also perform semantic 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 the videos which include cars, buses or trains, depending on the used ontological structure. Many different semantic similarity algorithms exist for conceptual ontology. However, none of the methods give the best result. Combinations of these methods like using edge counting and a corpus returns the highest accuracy. In our system, an edge counting method of Wu and Palmer is preferred [58]. This method is combined with the corpus method to get better results. Ontology structure has previously been used in the querying video systems. But these systems construct their own ontologies which need to be changed whenever the domain changes [33]. For the flexibility of the developed interface in this thesis, syntactic parsing with a general lexicon and domain-independent ontology search are used. So, when the domain and the video data entities change, no additional dictionary or algorithm processing is needed. To the best of our knowledge, there is no other video database system that facilitates natural language querying of the contents of videos in the same capacity as our video data model.

The reason we have chosen English instead of Turkish as the natural language in our thesis is the existence of a free and trusted link parser for English [31]. We are not aware of any such parser for Turkish. Another reason is the existence of a free source ontology called WordNet to retrieve the ontological structure of English words. The design of the system has been made independent of the language used in querying. Whenever there are tools for parsing Turkish sentences and tools providing the ontological structure of Turkish words the system could be adopted to provide query facility in Turkish.

The organization of the rest of the thesis is as follows: In Chapter 2, types of the video data models and the methods of retrieving data from video data models are discussed and the technologies used for the interfaces of video data and the NL interfaces are introduced. In Chapter 3, the video data model that is used as the basis of this study is explained by introducing the types of queries supported by this model. Our system maps the given queries in English into semantic representations of these queries. These semantic representations are built by the extraction module of the system. This extraction module and also the parsing technique of Link Parser are described in Chapter 4. An overview and implementation of conceptual ontology and the methods for similarity search are presented in Chapter 5. In Chapter 6, we explain how the semantic representations extracted from natural language processing module are mapped to the underlying the video data model. Finally, Chapter 7 presents the conclusions and future work.

CHAPTER 2

RETRIEVING DATA FROM VIDEO DATABASES

Video data is a combination of several multimedia data types like sound, image, text etc. Moreover, video data has spatial and temporal properties additional to semantic contents different from relational database [23, 24]. Since the data is unstructured and large in volume, it is difficult to manage, access, and compose video segments. All these factors require a video database system that is capable of storing and querying every property of any video element [11, 23, 24, 43]. Content-based querying is a kind of querying technique defined in various video data models [18, 37]. We studied on a video data model that supports content-based spatio-temporal queries. Spatio-temporal querying that enables to retrieve spatial and temporal properties of video objects in an integrated manner is an extended approach of content-based querying. Different techniques for query interfaces based on content-based querying are studied over video data models. Natural language interface is one of the major subjects among these techniques because of its flexibility. In this chapter, content-based querying techniques and interfaces are explained briefly. We also focus on the natural language interfaces over databases in this chapter.

2.1 Video Data Models

Depending on the recent advances in technology, research in multimedia databases has increased. Multimedia database management systems concern with the management of all the concerning data types including creation, storage, indexing and querying. MMDBS has the same functionalities of traditional database management systems like reliable data storage, data consistency, controlled mechanism of querying etc. The difficulty of multimedia databases is to control the variety of data types and enable flexible interface for the user [23]. In order to give a structured form to raw multimedia data, several data models have been proposed in the literature [1, 11, 13, 18, 24, 37].

Having different kinds of data types like sound, image, text; a video database must handle the storage and querying the properties of video elements. The main characteristics of video data:

- *Temporal:* Video data depends on time.
- *Spatial:* Depending on temporal properties, video is composed of image-like structures called *scenes*. In these scenes, the objects have spatial properties.
- *Size:* The video data has a large volume of size.
- *Variety of data types:* Video data is rich in information different from traditional databases because of the semantic content.
- *Unstructured:* Modeling of video data is very difficult in this concept; therefore fuzziness is a main problem. Because of the inexact boundaries of objects, relations between objects (different grades of membership can exist) and spatial properties in frames, there are uncertainties in video data models.

Having these characteristics, video data has a rich information source. Therefore, a variety of information may be retrieved from the same video data. Different modeling techniques on video data have been developed for these different application requirements.

Researchers have focused mainly on three modeling methods:

- *Video Segmentation Modeling:* A video stream is divided into video segments by using different techniques like histogram matching, algebraic operations etc. Whenever the histogram changes, a new video segment is created in histogram matching method [51]. The disadvantage of this modeling is its inflexibility.
- *Annotation-based Modeling:* Video is segmented semantically in this approach. These segments are annotated by keywords or attributes with free text, to help modeling data semantically. The annotations are used for keyword-based querying on video content. Hacid et al. proposed this method in his system which stored video annotations and associations between them to support querying video data [16].
- *Object-based Modeling:* By considering object-oriented approaches, this modeling technique focuses on semantic representations of video data. In the OVID system [37], video objects are the main subjects. Each video object has a unique identifier, frame numbers in which the object appears in and an attribute set. Another approach in this method is to index spatial and temporal properties of salient video objects in a video [13, 24, 52].

Indexing and retrieving video data are the subjects of the video data models. Each video data model uses different querying techniques. Researchers especially focus on querying the semantic content of the video data which is the subject of content-based querying [13, 18, 37].

2.2 Content-based Querying

The content of the video data includes objects, events, activities, and attributes of contents occurring in the video. In addition, the spatial and temporal properties of the objects may be of interest [11, 24].

The semantic content of the video can be queried in many different ways. Some example queries can be as follows:

- Elements contained like objects, events etc.: *Retrieve all frames where the teacher is speaking.*
- Spatial properties: *Retrieve all frames where the table is near the wall.*
- Temporal properties: *Retrieve all frames between 11.00 pm and 11.30 pm.*
- Spatio-temporal properties over video elements: *Retrieve all frames where the teacher is at the left of the table within the first 10 minutes of the video.*
- Audio data: *Retrieve the audio data where Petra Berger talks at the concert video.*

In video data models, various processing techniques are used. Some of them implemented video database query languages from scratch like a content-based video query language CVQL [25] and a declarative, rule-based, constraint query language used in [11]. And some of them extended the standard query language SQL [13, 37]. We can classify the existing video query languages as follows:

SQL-like querying: In some video systems like in OVID [37], SQL language is extended to deal with the spatio-temporal properties of video data. VideoSQL is the extended SQL language for querying video objects used in OVID. The query processing component of the system evaluates the intervals of video objects before each video object is considered in the execution of a query. However VideoSQL still can not retrieve temporal relationships on video objects with this extended language. In BilVideo [13], the database system defines special keywords for spatio-temporal properties. Fact-based rules are implemented as Prolog facts in a knowledge-base. When a textual query is entered, query processor groups the query into subqueries, and obtains the results by using knowledge-base facts for spatio-temporal properties. Spatial and trajectory queries can be processed by using a drawing-sketch tool, that the user can draw the paths or the Minimum Bounding Rectangles (MBRs) of objects. But in the textual query interface of BilVideo uses an extended SQL language. The disadvantage of this language is that the user needs to learn the details of an artificial language. Another SQL-like query language is used in AVIS [1]. But this language is also incapable of supporting spatio-temporal queries on events.

Specific video data query language: Users retrieve video data with spatial and temporal relationships for salient object by using a content-based logic video query language called CVQL [25]. Frame sequences satisfying a query predicate are retrieved by the help of the index structure proposed. The disadvantage of CVQL is that it does not allow topological and trajectory queries. TIGUKAT Query Language is another example for this type of querying. It is used for content-based spatial and temporal queries additional to 3D relation queries [29]. This querying has a difficulty in learning a specific query language as in the previous method.

Rule-based querying: Implemented in [11], a constraint rule-based query language using operators is proposed over a video data model which concerns with objects, intervals and their constraints in an extended active domain. Giving an example, a query is ruled in the form of:

$$Q :?q\ (\acute{s})$$

where q is referred to the query predicate, and ś is representing a tuple of constants and variable. This kind of querying has a disadvantage of difficulty in learning the rules.

The type of the querying technique determines the type of the query interface [26]. When the query interfaces are generalized, they can be categorized as follows:

- *Graphical Interfaces:* In visual query-formulation tools, pattern matching and trajectory drawing are two methods. Still-image representations of the video scenes can be used for QBIC-style querying. QBIC-style is a system that allows the user to query using sketches, layout or structural descriptions and other graphical information. By using several methods like sketching composed templates, color histograms etc. (used in WebSEEK and SWIM), the user can draw what he is expected in mind. In SWIM

(Figure 2.1), user composes the units of a canvas by specifying the visual attributes of these units. This composed key-frame of template is then matched with the frames. Color histogram method, which is based on modifying the color composition to be matched with scenes, is the method used in WebSEEk (Figure 2.2).

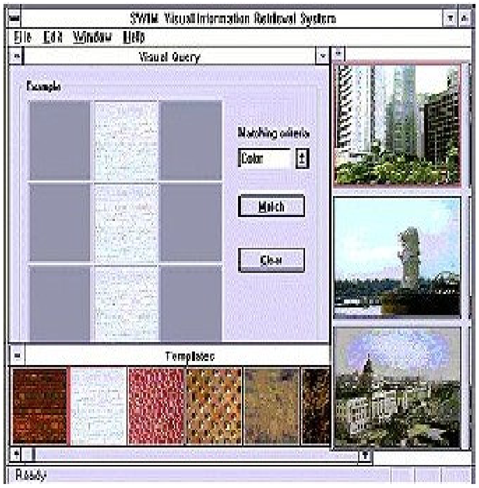


Figure 2.1 SWIM template composing

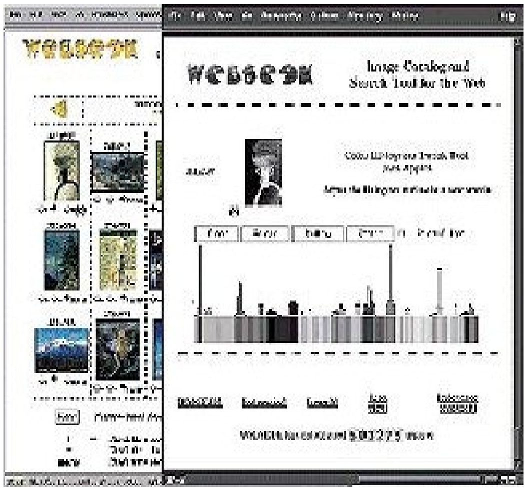


Figure 2.2 WebSEEk color histogram

With the help of sketch pads, user can draw the trajectory of the objects like in VideoQ (Figure 2.3). The system then searches the matching scenes to the trajectory in a ranked order. Another method in this line is to allow the user to select from examples (QBE-Query by Examples). These systems serve the user some example clips or shots. The user selects and asks for the clips that have same motions. In NeTra-V, the user selects from video sequences played and similar records are retrieved from the video database also by considering the color and texture.

- *Textual Interfaces :* In textual query formulation tools, user can type the queries formulating terms with boolean operators like in VISION (Figure 2.4). These terms are matched with the indexes of the video clips automatically. Another method of textual interfaces is to enter the queries with qualifiers and listed keywords. These qualifiers and keywords are matched with the indexed video keywords defined in the database [43], similar to VISION. SWIM and WebSEEk use a category/hierarchy of video database in its interface and allow the user to explore between these like in an explorer window to help query-formulation from indexes. WebSEEk is a Web-based catalogue system that retrieves context-based image/video files from the Web [27]. In some interfaces, list of elements and query index terms are listed for the user to choose to formulate the query at the background like in VideoSTAR (Figure 2.5) and in [23, 24].

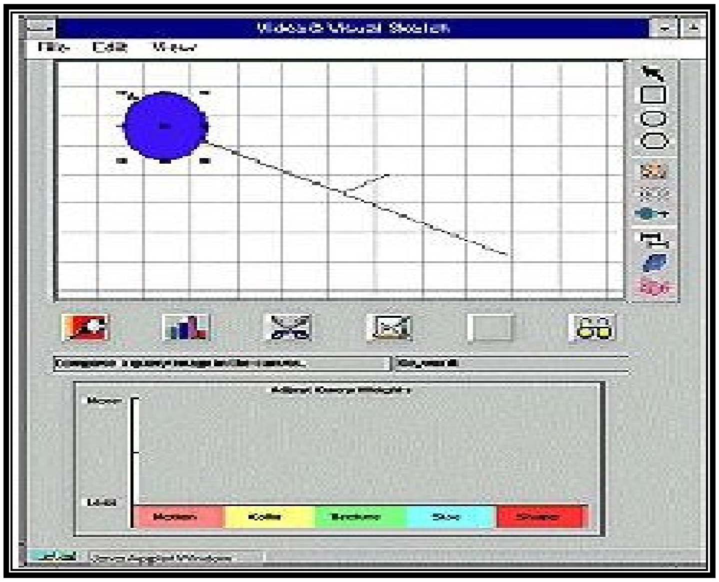
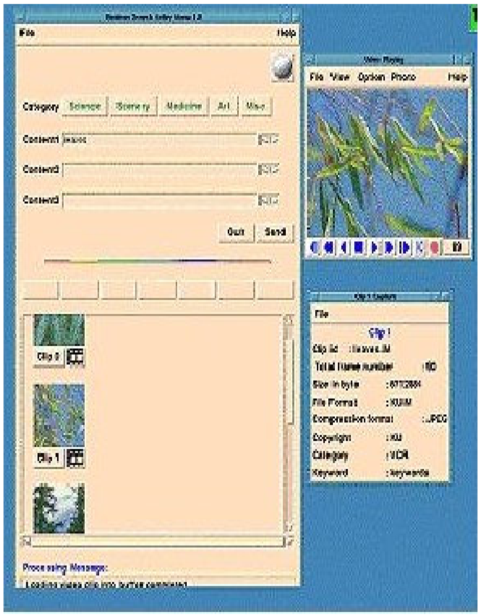


Figure 2.3 VideoQ trajectory drawing

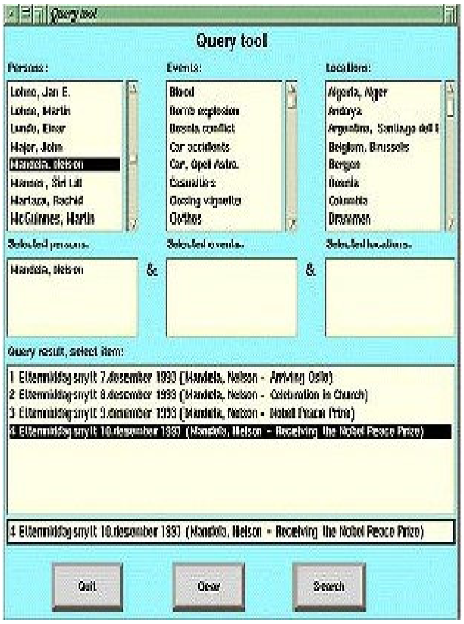
VideoSTAR, which is a generic video database, is especially used for searching TV broadcasting/news and documenting professional archives [27]. The most flexible method in this category is to use natural language processing at the interface like in Informedia [19] and VideoQ [10]. In this system, the user does not need to learn any artificial query language. He just uses his own sentences for the query. In Informedia [19], the terms parsed from the query are tried to be matched with the keyframes assigned to each video clip. In the next section natural language query interface will be described in detail.



Clip 1
Clip Id : leaves.M
Total frame number : 60
Size in byte :
File Format :
Compression format : JPEG
Copyright :
Category :
Keyword : keywords

Processing Message:
Loading video clip into buffer completed

Figure 2.4 VISION interface



Persons:	Events:	Locations:
Lohne, Jan E.	Blood	Algeria, Alger
Lohne, Martin	Bomb explosion	Andorya
Lundo, Einar	Bosnia conflict	Argentina, Santiago del E
Major, John	Car accidents	Belgium, Brussels
Mandela, Nelson	Car, Opel Astra.	Bergen
Manus, Siri Lill	Casualties	Bosnia
Martaza, Rachid	Closing vignette	Colombia
McGuinnes, Martin	Clothes	Drammen

Selected persons: Mandela, Nelson

Query result, select item:

1 Ettermiddagsnytt 7.desember 1993 (Mandela, Nelson - Arriving Oslo)
2 Ettermiddagsnytt 8.desember 1993 (Mandela, Nelson - Celebration in Church)
3 Ettermiddagsnytt 9.desember 1993 (Mandela, Nelson - Nobel Peace Prize)
4 Ettermiddagsnytt 10.desember 1993 (Mandela, Nelson - Receiving the Nobel Peace Prize)

4 Ettermiddagsnytt 10.desember 1993 (Mandela, Nelson - Receiving the Nobel Peace Prize)

Figure 2.5 VideoSTAR interface

2.3 Natural Language Interfaces

There are various methods for retrieving the video data. Each method has its own advantages and disadvantages. In this study, natural language interface is used for querying 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 great advantage of natural language processing (NLP) [5]. NLP sometimes is the most flexible way of expressing queries over complex data models. On the other hand, there are still some disadvantages that users are limited by the domain and by the capabilities of parsers and also the linguistic and conceptual definitions are not clear. Therefore 100% accuracy cannot be achieved. However in recent studies, NLP techniques have improved considerably and also it is possible to obtain approximately 90% accuracy in query results.

2.3.1 Natural Language Interfaces Over Databases

Early studies of natural language query processing depended on pattern-matching techniques. These were 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 types of queries and they are executed after the queries. 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 consists the word 'capital' before a country name*". Then the query "*Retrieve the capital of Italy?"* will answer "*Rome"* as the result. However, since the results of this technique were not satisfactory, new techniques have been developed.

Another NLP method is used in the system LUNAR [56]. It is the 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 mapping from parse tree to the database query. Moreover, it is difficult to decide the mapping rules from the parse tree to the language (e.g. SQL) that the database uses.

Semantic grammar systems also use the parse tree as in the system LADDER [5]. However its nodes do not correspond to syntactic categories. The disadvantage of the method is that semantic approach needs a specific knowledge domain, and it is quite difficult to adapt the system to another 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 one of the examples for this type. 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 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.

Database and domain dependence is an important subject to consider for every method. Minimum dependency should be chosen to enable a more flexible system when deciding on a technique to implement a natural language interface over a data model.

2.3.2 Natural Language Techniques over Video Databases

As a 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 like in SPOT and Informedia [19, 21, 33].

The video data system, SPOT [21], queries about moving objects in surveillance videos. It uses the natural language understanding in the form of START (a question-answering system), which has an annotation based natural language technique. In the knowledge base, annotations are stored which are English phrases or sentences. The phrases are used for describing the question types and the information segments. Queries are syntactically parsed to match with these annotations. Further from syntactical level parsing; ontologies, structural transformations, synonym/hyponymy pairs are also processed in the matching phase. 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, tracks are the main basic unit of data, which traces the motion of a single object in a 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 has some disadvantages too. For instance it is incapable of capturing high-level semantics and annotations need extra labor.

In [33], media data in the query is extracted with natural language into the *description data* by using a matcher tool (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 in natural language, same descriptions can have different semantics. Therefore, approximate matching is considered in the system. In order to perform the approximate matching, semantic network model is used. This model consists of networks of nodes. When the query is parsed, the semantic representation is translated into a semantic network tree which the major nodes are nouns and the actions in the query. 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 following the hierarchies. Weights are used in the hierarchy trees to enable better matching results. The disadvantage of the system is to decide the weights and normalization of the statements in natural language.

In the systems that use natural language interfaces, matching the query with the video data in the database is an important problem. To solve this problem, main terminology is on entities that are shown in video frames/shots and the activities that these entities do in the given time interval and place. So when queries are parsed, the first aim is to extract the entities that occur in the query and match them with entities in the database. But sometimes, an exact matching can not be obtained from the query and the knowledge base. For example, the user can ask for a *car*; however in video database there may not be any *car* entity, but instead *Mercedes* and *Fiat* exists as entities. In order not to reply with an empty result set, ontology based querying is used after the parsing phase [3, 22]. Similarity between entities in the database and parsed entities from query is evaluated by using an is-a hierarchy tree. 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, the method for implementing a natural language interface over a video data consists of syntactically parsing the query and using ontology to find better results and exact matches. In this thesis, a natural language interface over a content-based video data is implemented for spatio-temporal querying. The system becomes more flexible than using a rule-based or extended video language with the advantage of NLP. Whenever a query is parsed, the semantic similarities are evaluated by using ontology (WordNet). At least similar or related results are returned to the user by the help of the ontology. This is also an advantage among the other NLP interfaces. Some NLP interfaces use keyword-matching system like Informedia [19]. These systems have the disadvantage of missing the detailed and more realistic queries like ‘rainy weather’. However our system can distinguish the semantic details. This advantage enables the system being extendible for compound spatio-temporal queries and other fuzzy elements which most systems do not have this ability. Unlike OVID [37] and similar systems, our NLP interface enables all facilities of spatio-temporal properties supported by the underlying video data model. In the next chapter, the video data model is described to give an idea of the query types to be parsed.

CHAPTER 3

THE VIDEO DATA MODEL AND QUERY TYPES

This chapter introduces the video data model and the query types supported by this model. An object-based model enabling fuzzy spatio-temporal queries is used in the system. In the previous studies over this data model [23, 24], a basic graphical user interface is implemented for querying. This query interface is implemented depending on the query types supported. An NLP querying interface model is decided to be implemented for the types of the query in this thesis, as there is the lack of flexibility on this interface.

3.1 The Video Data Model

The data model in the system is an extension of AVIS (Advanced Video Information System) [1]. Our system is an object-based data model which focuses on entities in the video. The video clip is divided into time-based partitions called *frames*. Entities are objects, activities and events appearing in the video frames. In order to define the temporal properties of entities, each entity has a frame sequence set attached to it. By using some indexing structures on these entities, information is retrieved and correlations are made between entities during the query processing.

The entities defined in the model are:

- *Object:* Video objects are defined as visible entities in video frames. e.g. *cat, Seda, football,* etc.

- *Activity*: Activities are the subjects of the frame sequences. They are grammatically verbal clauses like *playing football, running,* etc.
- *Event:* Activities and one or more objects define events like *John playing football*. Objects are defined as actors and the activity is defined as the role of the actors in the event definition.

Frame sequences are used as temporal data of the video. Therefore, they can be used to evaluate the time intervals of the video. They 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 frame sequences, in which they occur. These definitions are combined in the *association map* [24]. The model provides semantic entities to represent occurrences in a video, frame sequences to identify the occurrence time intervals for entities, and association maps to combine entities with time intervals. In this map, horizontal line segments correspond to occurrence of an entity represented on y-axis during the frame intervals represented on x-axis as shown in Figure 3.1.

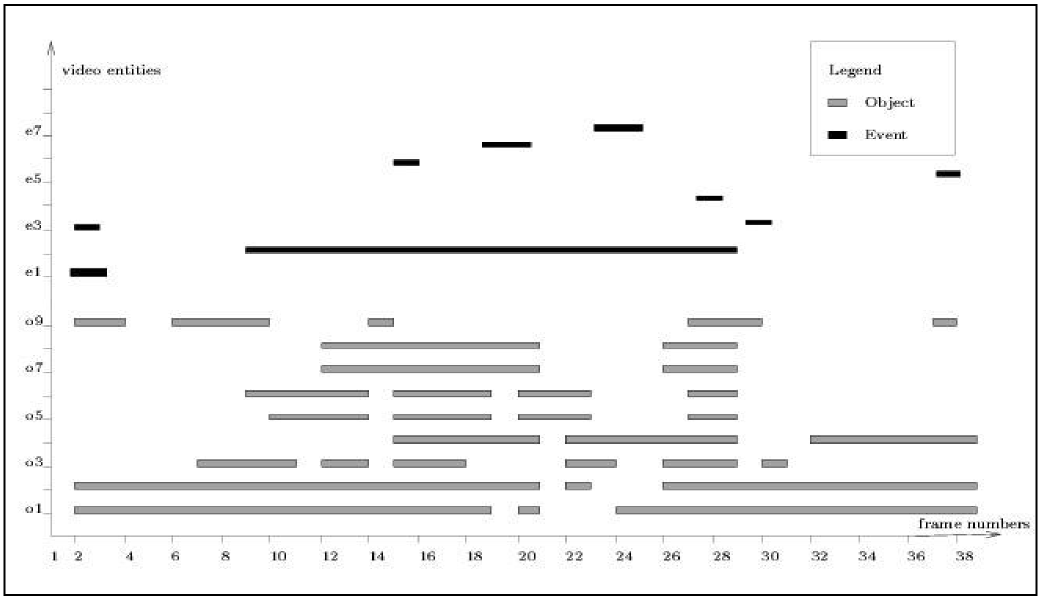


Figure 3.1 A sample association map

Frame segment tree is a kind of tree structure that represents the association map lines. Each node in the tree represents a frame sequence like [x, y) starting at frame x and including all frames up to y but not including y. Each node is associated with numbers corresponding to the entities of objects and events that appear in the frame intervals represented by this node. Whenever an entity exceeds the frame interval of a node, it is represented more than one node.

The model designs arrays for objects, events and activities in which each entity in the video data is loaded depending on the type of the entity. Each element of the array structures is associated with the pointers to the nodes of segment tree. These nodes are the ones that this element occurs in. By this way, the temporal properties of the entities are handled.

Frames can be queried by giving the object/event or activity in the type of elementary queries. The pointer defined in the identities' array is traced through the segment tree to get the frame sequences. Also giving the frame sequence from the video clip, the objects or the activities shown can be queried by implementing the same algorithm. Conjunctive and compound queries can also be implemented. So the outputs of the queries can be intersected (conjunctive) or the output of one query can be the input of another query in one query transaction.

Spatial properties contain the location information of an object in the video frame [23]. To represent spatial relationships between two objects in a video is more difficult than in image for the fact that video data has time-dependent properties. In this model, a two-dimensional coordinate system is used for the spatial properties. The most preferred method to define the location of an object is to use an MBR (minimum bounding rectangles) which is an imaginary rectangle that bounds an object's area 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.

In this model, 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 fuzziness into the model [24].

We use a threshold value as a fuzzy membership value between [0, 1] for any spatial relationship. This value indicates the satisfaction degree of the spatial positions of two objects for a given spatial relationship. The angle between the centers of the rectangles is evaluated to find the threshold value. This calculation enables to perform fuzzy queries in the model. In the data model, four main spatial relations are studied as left, right, up and down. The representation LEFT (A, B, 0.7) means that object A is at the left of object B with a membership value of %70. That is, object A is not exactly at the left of B, but it is approximately at the left of B (see Figure 3.2).

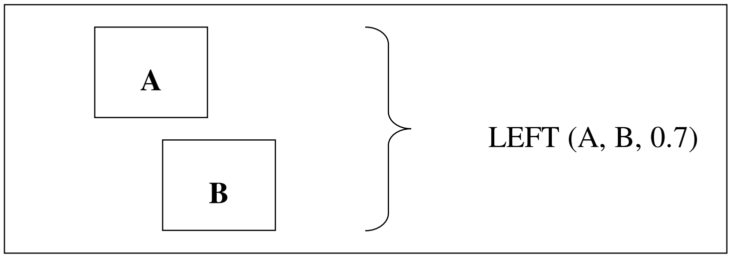


Figure 3.2 Example of a fuzzy LEFT relationship

3.2 Query Processing of the Video Data Model

This data model supports many different query types. In addition to basic querying of objects, activities and events; it is also possible to query spatial properties of objects, spatio-temporal relationships between objects and trajectory of objects.

3.2.1 Query Types Supported by the Data Model

The supported query types are as follows:

- Elementary Object Queries: Given an object one may query the frame list. *E.g. Find all frames where Aysu appears.*

- *Elementary Activity Queries:* Given an activity one may query the frame list.

  E.g. Find all frames in which somebody plays football.
- *Elementary Event Queries:* Given an event one may query the frame list.

  E.g. Find all frames in which the cat is catching the mouse.
- *Object Occurrence Queries:* Given an interval, one may query the objects in this interval. It is a kind of temporal query. The intervals are given in minutes depending on the duration of the video clip.

  E.g. Find all objects appeared during the last 10 minutes of the film.
- *Activity Type Occurrence Queries:* Given an interval, one may query the activities in this interval.

  E.g. Find all activities performed in the first 5 minutes.
- *Event Occurrence Queries:* Given an interval, one may query the events in this interval.

  E.g. Retrieve all events appeared during the last 30 minutes of the video.
- *Spatial Relationship Queries:* Given two objects and relation with the fuzziness value, one may query the frame list.

  E.g. Find all frames where the chairman is at the left of the rector.
- *Regional (Spatial) Queries:* It has two kinds:
  - Given an object and the interval, one may query the regions.

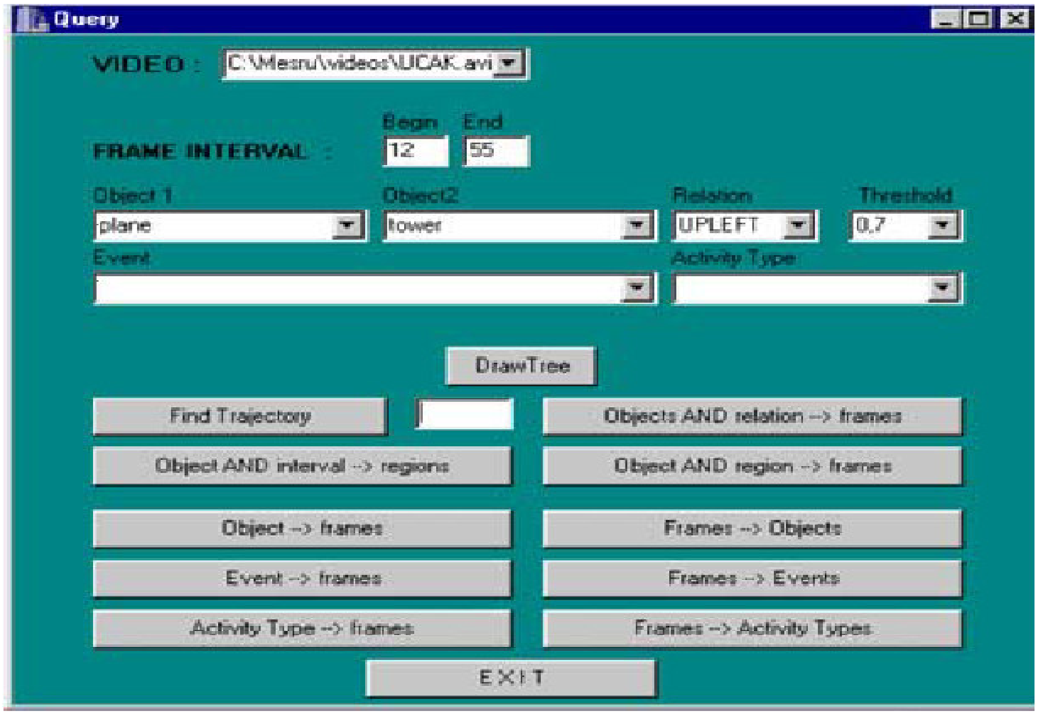
    E.g. Give the regions where the plane is seen during the last 10 minutes.
  - Given an object and the region, one may query the frames.

    E.g. List all frames where the plane is at the right of the screen.
- *Trajectory Queries:* Consecutive frames in the video including an object's route can be queried. Since a trajectory is formed of a set of continuous (interval, region) pairs, it is queried both the time interval and the spatial location of an object in this type of query. The consecutive regions in the pair set are sorted as any region is the neighbor of the previous region. In the same way, the consecutive intervals in the pair set are sorted as any interval is the successors of the previous one.

E.g. Show the trajectory of the ball going from left to the right of the screen.

3.2.2 Previous Query Interface of the Video Data Model

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 was 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 as seen in Figure 3.3. The results of the query are shown in snapshots. Using the drop-down lists, results are shown in video clip or listed in list-type windows.



VIDEO : C:\Mesru\videos\UCAK.avi

FRAME INTERVAL : Begin 12 End 55

Object 1: plane

Object2: tower

Relation: UPLEFT

Threshold: 0,7

Event:

Activity Type:

Figure 3.3 Previous query interface of the video data model

But this interface has been very restricted for the user and also for the project itself. Because whenever a new query type is added, a new button and also new text boxes for entries should be coded. Another disadvantage is that if the database is large, to find an object identity from the drop-down lists is a handicap. There is no flexibility to use ontology either. Only the objects defined in the database are shown to the user. Therefore no generalization or specification can be applied to the objects.

3.3 Queries with Natural Language

In order to have a more flexible querying environment, we have decided to use a natural language interface. The user can ask the query with his own words with a natural language interface. It is the most flexible method among the other query techniques. The aim in this thesis is to implement a natural language interface over the video data model explained in Section 3.1. The query types given in Section 3.2 are asked by the user in English. The queries can be in the format of questions by using wh-words like when, who etc. or can be imperatives starting with *retrieve, find, give* etc. The most known disadvantage of a natural language interface is being unbounded. It means that so many ways to form questions can be found. But to enable the robustness and accuracy, the user should be restricted by some question patterns. In this implementation, only a restricted set of queries are allowed to be asked with defined parameters.

Table 3.1 illustrates the query types in natural language. First column of the table shows the query types supported by the model. In the second column semantic representation for each query type is presented. Each representation has a name, one or more input parameters and a return value. An example for each query type can be seen at the last column of the table.

Table 3.1 Query examples in NLP and their semantic representations

Query Types	Semantic Representations of Queries	Examples
Elementary Object Queries	RetrieveObj (objA) : *frame_list*	Retrieve all frames in which Bush is seen.
Elementary Activity Type Queries	RetrieveAct (actA) : *frame_list*	Find all frames in which somebody plays football
Elementary Event Queries	RetrieveEvnt (evtA) : *frame_list*	Show all frames in which Albert kills a policeman
Object Occurrence Queries	RetrieveIntObj (intervalA) : *object_list*	Show all objects present in the last 5 minutes in the clip.
Activity Type Occurrence Queries	RetrieveIntAct (intervalA) : *activity_list*	Retrieve activities performed in the first 20 minutes.
Event Occurrence Queries	RetrieveIntEvt (intervalA) : *events_list*	Find all events performed in the last 10 minutes
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
Regional(Frame) Queries	RetrieveObjReg (objA, region) : *frame_list*	Show all frames where Bill is seen at the upper left of the screen
Regional(Interval) Queries	RetrieveObjInt (objA, intervalA) : *region_list*	Find the regions where the ball is seen during the last 10 minutes.
Trajectory Queries	TrajectoryReg(objA, start_region, end_region) : *frame_sequence*	Show the trajectory of a ball that moves from the left to the center.

Some issues are considered as seen from the examples, when a natural language interface is implemented:

- For interval, spatial relationship, region and activity word groups in the queries, some defined structures are expected. For example; all intervals are expected as *the last 10 minutes*, *the last 5 minutes* etc. Activities are distinguished from the events by the words like *somebody, something, anybody* etc. which means that the actor is not defined.
- The query should be formed as a single sentence. Elliptical queries (Queries that would follow each other and use common pronouns) are the subject of the future study.
- The query must be syntactically and grammatically true. If not, no results will be shown.

When the query sentence is processed by the system, semantic representations of queries are tried to be formed to be mapped to the video data model. So for each type of the query, a semantic representation is defined. The representations have a defined name for each type. The parameters are fixed in number and their concept is also defined. The query is syntactically parsed and parameters are extracted. The output shown after ‘:’ is the list that the user asked to be retrieved. This output type is defined depending on the type of the query.

CHAPTER 4

MAPPING QUERIES TO SEMANTIC REPRESENTATIONS

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, an information extraction module and a conceptual ontology. 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 4.1. In the rest of this chapter, the querying system using natural language is explained in detail. Semantic representations, the parsing technique to get representations, information extraction depending on query types and mapping to semantic representations are the subjects of the chapter.

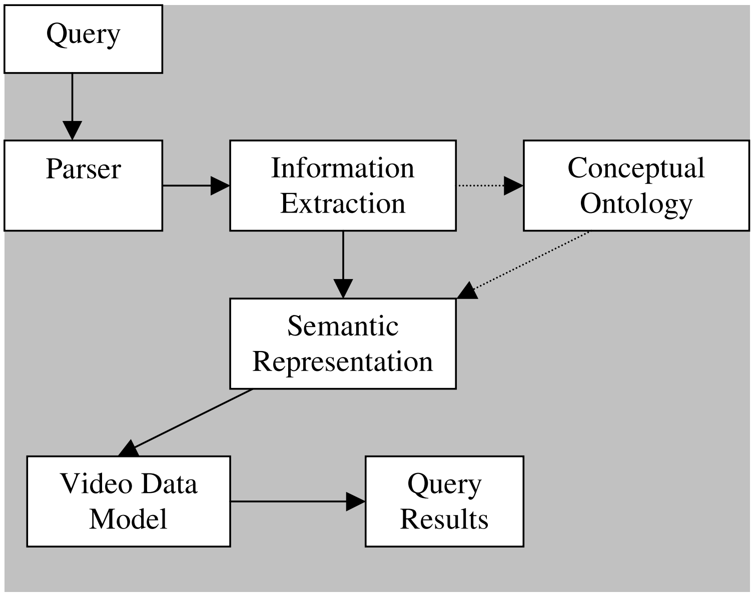


Figure 4.1 System Design

4.1 Semantic Representations of Queries

The query types supported by the video data model are all pre-defined as described in Chapter 3. The problem while implementing the interface with NL is to extract the query elements from the sentence and get these elements into an available format to map them into the video data model. Therefore, this format must match with a query type defined in the video data. Since every query type has a format as seen in the second column of Table 4.1, the query can be extracted to obtain semantic representation similar to these formats. This will help the semantic representations to be matched with the video data query representations.

When the user enters a query in NL, the first step is to find the query type in order to get the parameters of the query. The semantic representations are in the following format:

Table 4.1 Semantic representations of query types and query sentences

Query Types	**Semantic Representations of Queries**	**Examples**	**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).
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(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]

type_of_function (*parameter1, parameter2, ...*) *: return_value*

type_of_function is the unique value for the name of the representation. The number of parameters depends on the *type_of_function.* For each type of the query, the number of parameters is pre-defined. *Return_value* also depends on *type_of_function*. It represents what is asked to be returned (e.g. frames, intervals, objects, trajectory etc.).

After the query is entered, the parts are parsed to be included in semantic representations. Therefore, the parsing algorithm is focused on only the candidate elements for the representation. The query type can be defined after the parameters and return-value of the query is extracted. So the focus is on the parameters of the semantic representation. In Table 4.1, semantic representations of the supported query types are shown with examples.

Every query should include at least an object, an event, or an activity as in the structure of the video data model [15]. Object and activity are basic particles that sometimes form an event. If the basic particles are found initially, then the parameters that include one or more basic particles can be expanded. Since an event contains an activity and one or more objects, first of all, the activity and activity objects should be found for the event to be formed.

4.1.1 Objects

The semantic representation of an object is:

Object (object_name, attribute1, attribute2 ...)

Object is the predicate name used in the semantic representation of the query. The parameter *object_name* is the name of the object in the query like *cat*, *Istanbul*, *Elton John* etc. The object can have one or more attributes like *red*, *thin* etc. In this implementation, because that the video data model does not support attributes for objects, they will be null when the query is extracted. Also as a future work, the number of the attributes will not be limited.

When the query is parsed, objects are nouns; attributes are adjectives of these nouns. The important point is, not every noun is an object. Depending on the parsing technique, some rules are implemented on the sentence to get the right noun as the object. This technique will be explained in Section 4.2.

For instance; consider the query:

Retrieve all frames where the cat is seen in the video. (Frames, cat and video are all nouns; but the only object here is the cat.)

4.1.2 Activities

The semantic representation of an activity is:

Activity (activity_name)

Activities are subjects in the video frames and verbs in the queries. They are also basic terms. *Activity* is the predicate name used in semantic representation of the query. *activity_name* is the activity verb itself. Similar to objects, not every verb in the query sentence means an activity. It depends on the parsing technique to determine whether a verb is an activity or not. An example query could be :

Retrieve all frames where somebody is playing football.

In the sentence above, there are two verbs: *retrieve* and *playing football*, the parser extracts only the verb *playing football* as the activity of the query.

4.1.3 Events

Events are not atomic, because every event has an activity and the actors of that activity as parameters. The semantic representation of an event is:

Event (activity, object1, object2 ...)

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 before the event extraction. Therefore, if the event is *Albert kills a policeman,* first of all the following predicates are extracted:

Activity_A (*kills*)

Object_A (*Albert, NULL, NULL*)

Object_B (*policeman, NULL, NULL*)

Then, Event_A (*Activity_A, Object_A, Object_B*) is extracted as the full semantic representation of the event.

4.1.4 Spatial and Temporal Semantic Representations

There are other kinds of semantic representations for spatial and temporal properties in the query. Some of them are basic structures using coordinates and minutes, and some of them are relations between any two objects.

Regional queries are the type of spatial queries that ask for a given object in a given region. They include some rectangle coordinates to describe a region as the spatial property in a frame. During information extraction, the phrases representing these rectangles must be converted into two dimensional coordinates in order to map them into the functions of our data model. Thus, regional semantic representation is:

Region (x1, y1, x2, y2)

Region is the predicate name that will be 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 the regional rectangle (Figure 4.2).

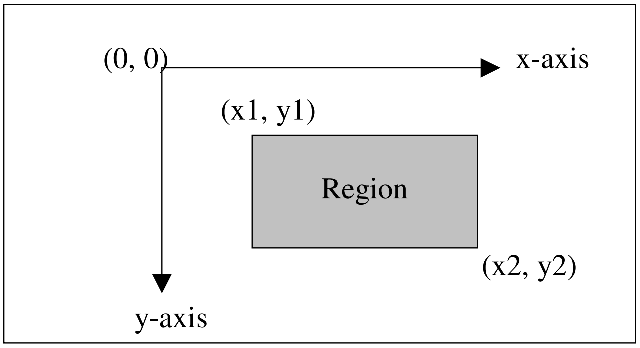


Figure 4.2 Coordinate representation of a region

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

Interval (start, end)

Interval is the predicate name that will be used in the query semantic representation. *start* and *end* are the bounding frames of the interval in the type of minutes. They are evaluated depending on the duration of the video.

All the representations presented so far are basic representations like an object and an activity. This means they can be used in a more general representation – in a query or an event-.

Spatial relations are extracted as predicates representing the spatial relationships, and the extracted objects involved in the spatial relations. Therefore, they differ from temporal and regional queries by not being basic terms for the reason that they involve extracted objects. 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 the representation, *threshold* value is used to specify the fuzziness in the spatial relations. It must be a value less than or equal to 1. These predicates are pre-defined in the video data model, but for future studies about the video data model, the relationship directions can be extended; relatively the predicates can be changed.

The result of the extracted query will be one of these semantic representations. After finding the basic representations, the query representations will be formed as seen in Table 4.1. Each query in Table 4.1 has a different semantic representation, and they have a different set of parameters in their semantic representations. Therefore, 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. To get semantic representations for the query, a parsing algorithm is used to find the syntactic structure of the query, and the information extraction module extracts the semantic representation of the query from its syntactic structure.

4.2 Parsing Queries

In order to extract information from the user query, a syntactic parser is needed. In the thesis, only specific kinds of word groups (like objects, activities, start of the interval etc.) are needed to obtain the semantic representations explained above. For this reason, a light parser algorithm is decided to be used for parsing. A light parser does not need to find the whole detailed parse tree of the query sentence. Instead of writing a parser algorithm from scratch, a known light parser is decided to be used not to lose time and digress from the purpose.

Light parsing techniques are used when it is sufficient to parse only groups of words. These are summarized in the following:

- *Chunk parsing*

This technique, which is introduced by Abney in 1991, has advantages of speed, less memory and robustness than full parsers [6, 44]. The logic depends on obtaining non-recursive (never has a phrase of the same category) chunks by tagging and tokenization instead of using context-free rules. It is mostly used in information extraction and message understanding [32]. Chunk parser builds single-level trees that are composed of continuous and non-overlapped groups of chunks. Then, these chunks are combined to form chunk structures as two-level trees also including unchunked tokens. In Figure 4.3, a noun phrase chunk (NP) is combined by the parser to form the chunk structures of S.

NP: <a> <yellow> <taxi> → *a chunk representing an NP*

(S: (NP: <He>)

<drives>

(NP: <a> <yellow> <taxi>) → *a chunk structure capturing the NP in a sentence*

Figure 4.3 NP chunks are parsed in the sentence

To decide for chunks, the parser needs tags. In [6, 32], chunk parser examples NLTK and SCP are explained by enabling some specific methods for the parser like *chunk, unchunk, show_tree etc.* By writing only the specific phrase rules like for NP, VP only these phrases are chunked. The whole parse tree is not needed for this parsing technique.

- *Shallow parsing*

Similar to chunk parsing, the goal is to obtain an efficient and robust bracketing technique to get partial analysis of the syntactic structures in the text. It uses statistical techniques to find the best bracketing [35] and most alike information as in the reference corpus [53]. Described in [35], open/close predictor method can be used to obtain specific phrases like NP, SV (subject-verb). For each word in the sentence, the phrase boundaries are evaluated and probable combinations are evaluated to get the highest result. POS (part of speech) information and relational features also affect the results like in chunking.

- *Link parsing*

The method used in this thesis is link parsing because of more accurate results and easier implementation. This parser does not need a corpus and POS tags for the input sentence. Another advantage of this parser is to have the ability to get the grammatical relations between word groups. They are needed when extracting semantic representations. In the next section, link parser is explained in detail.

4.2.1 Link Parser

Link parser is a kind of light parser which parses one English sentence at a time. It has an API that has functions to use the parser easily. When the sentence is given as an input to the parser, sentence is parsed with *linkages* using its grammar and its own word dictionary.

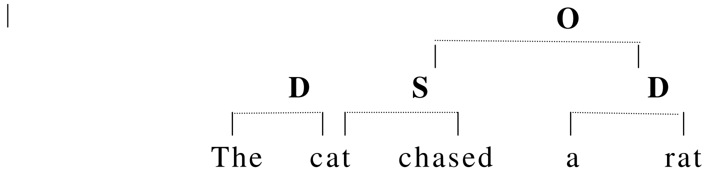
It has a basis depending on three important rules [31, 49].

- Satisfaction: For every word, there is a linking rule that satisfies its requirements.
- Planarity: Links, which connect two words, never cross in a sentence.
- Connectivity: Each word in the sentence must be connected at least by one link.

Link types, which are defined in the link parser *dictionary*, have grammatical rules to connect two different words. The parsed sentence can be described as a tokenized input string by links which are obtained by the sentence splitter. A word can be the right or left *connector* of a link type like in the example below:

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

Ds is a link type. It connects determiners to nouns. 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.



As seen from the example; no two links crosses and no words are left as unlinked. There are 60,000 words in the parser's dictionary and it has an API for the basic operations which are also used in the system. The links are represented in capitals like *A (connects pre-noun adjectives to following nouns), D (connects determiners to nouns)*. The small letters near the representation of link types are the links' attributes like *Ps, Ds (s means singular)*. The link types used in the implementation are listed in Appendix A.

With the help of the link grammar, only needed groups of words can be parsed by selecting proper links or link series. Another advantage of the link parser is to determine the part of speech tags of the words (as .n, .v etc.) in the query sentence as shown in Figure 4.4 to help obtaining link rules. This advantage also helps to distinguish words in information extraction, especially in proper nouns. The parsing algorithm of link parsing uses the link type rules in a recursive manner to search for every linkage over each word.

To solve the problem of finding out specific word groups from query, the query is parsed by the link parser. These word structures are extracted by information extraction module of the system by scanning and tracking special link types obtained after parsing.

```
                         +------------MXs-------------+
                         +----Bs---+    +-----Xd-----+
  +---Ss--+--Pa--+---MVs--+--Cs--+-SFst+-Ost-+--R--+-I-+    |     +---G--+-Xc+
  |       |      |        |      |     |     |     |   |    |     |      |   |
grammar.n is.v useless.a because there is.v nothing to say.v -- Gertrude Stein .
```

Figure 4.4 A parsed sentence with link parser with POS tags

4.2.2 Information Extraction Module

Parsing specific kinds of words from the query sentence is the information extraction module of the system. This module also forms the semantic representation of the query from the output of the parser.

A similar technique is also used in crime scene reconstruction [14] which has been adopted from information extraction methodology used in SOCIS [39]. SOCIS is the scene of crime information system. The captions describing the crime scene are extracted to get relational facts in the format of *argumentA RELATION argumentB*. These facts are used for indexing crime scene photos in an application domain. The relations are especially spatial prepositions used in the sentence. To retrieve related crime photos depending on the user query, query is extracted as the format given above. These triplets are then matched with the ones in the indexed triplets. Depending on SOCIS, another system concerns with scene generation [14]. This system takes crime scene sentences as input and extracts the sentences for scene generation. When the sentence is entered to the system, it is parsed by Link Parser. By the help of the rules defined on linkages information extraction is implemented.

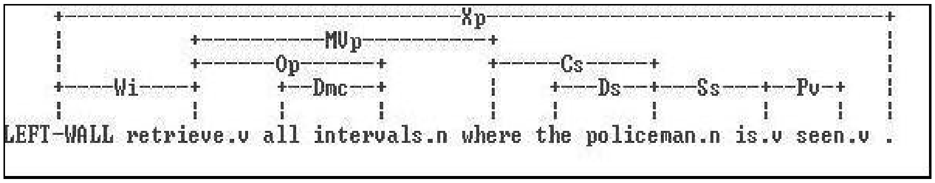
In our system, extraction module depends on a rule-based extraction defined on linkages of link parser. The aim is to extract word groups in the parsed query to map to the semantic representations defined for *objects*, *activities, intervals, regions* and *spatial relations*. After extracting the basic terms, the query's whole representation 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 to be extracted, one or more rules are written (in Appendix B). 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 instance, when the query is entered as:

Retrieve all intervals where the policeman is seen.

The output of the link parser is as follows:



```
    +---------------------------------Xp-------------------------------+
    |          +------------MVp--------+                               |
    |          +------Op------+        +------Cs----+                  |
    +----Wi----+      +--Dmc--+        |    +---Ds--+---Ss---+--Pv-+   |
    |          |      |       |        |    |       |        |     |   |
LEFT-WALL retrieve.v all intervals.n where the policeman.n is.v seen.v .
```

To understand the type of the query, any *Op* or *Ce* link is searched. The right-end of the link is the return value of the semantic representation. In this query, *intervals* is the return value. Also it shows that this query is a kind of interval query. Therefore an object, an activity or an event will be the parameter of the semantic representation depending on this query type definition. As there is a *Cs* link at the output, there is an object or an activity in the query. An activity needs a path as *Ss+Pg*. However, only a *Cs* link exists in this sentence. Therefore depending on the rules defined, the right-end of the link is the object in the query. While the semantic representation of this query type has the form as:

RetrieveIntervalofObj (objA) : interval_list

The resulting semantic representation is formed as:

RetrieveIntervalofObj (policeman) : intervals

In the extraction algorithm, every link and every word in the query gets an index and label after parsing. Tracing all links in the parsed query, specific link types (in Appendix A) are searched. The rules that are written for the specific word groups are applied tracing this link path whenever these links are found. The words that are connected throughout this path form the word group. All these operations are done by the help of indexes of words and links. By the help of the link parser data types, the right and left connectors of a link can be learned and labels of the words on the connectors can be retrieved.

For instance, the following rule is one of the rules that are used to find an activity:

- *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)*

In the Appendix A and B, all the rules and link types used are given.

4.3 Mapping to Semantic Representations

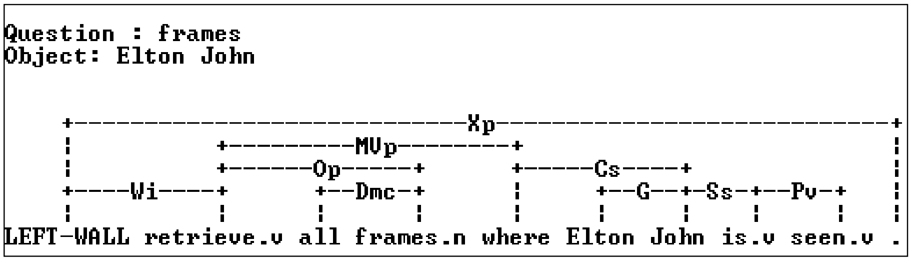
For each query, first the query type is extracted from the parsed query. The question word and the keyword that is asked obtain the return value of the query.

E.g. *Retrieve all intervals where somebody talks.*

In this sentence above, the word *intervals* is the keyword for the return value of the query's semantic representation. It also gives a clue about the type of the query. If only an object or an activity or an event is extracted, this means that the query is an interval type.

After extracting all atomic word groups like objects, relations, intervals, regions etc., the semantic representation of the whole query is constructed. For example, if the query is a trajectory query, an object, a starting region, and an ending region should be found. The only query type that involves the semantic representations of two regions and an object is the trajectory query. Therefore, the atomic representations are mapped to the representations of trajectory query representations including the return value extracted from the query. (Semantic representation of the trajectory queries: TrajectoryReg (objA, start_region, end_region): *frame_sequence).* In Appendix C, some query examples and their semantic representations are given with the output of the parser.

As another example consider Figure 4.5. Op linkage helps us to determine the return type of the query as *frames.* Then any atomic representation is searched tracing the linkage paths. An object, *Elton John*, is found when the Cs link is traced. The tracing process is finished, when there are no more atomic representation can be found. So depending on the keyword *frames* and only one object representation, the query type is decided to be as an Elementary Object Query.



```
Question : frames
Object: Elton John

    +-----------------------------Xp---------------------------+
    ¦          +---------MVp--------+                          ¦
    ¦          +------Op-----+      +-----Cs----+              ¦
    +----Wi----+      +--Dmc-+      ¦     +--G--+-Ss-+--Pv-+   ¦
    ¦          ¦      ¦      ¦      ¦     ¦     ¦    ¦     ¦   ¦
LEFT-WALL retrieve.v all frames.n where Elton John is.v seen.v .
```

Figure 4.5 Information extraction example from a query

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 a word phrase in a given query (such as 1 versus *one*), but it needs to be a numerical value in the data model. 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. But in video data model regions are represented as two dimensional coordinates. To map this data with the video data model, these areas should be converted into the coordinates. Thus, the regions can be represented as rectangles.

The screen is thought to be divided into five regions as upper-left, upper-left, down-left, down-right (Figure 4.6).

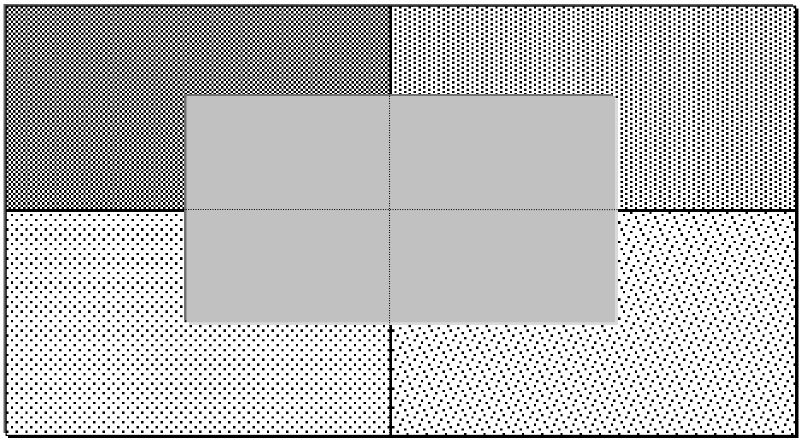


Figure 4.6 Screen is divided into five rectangles as regions

Depending on the area in the 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. By taking the initial screen coordinates at the beginning, the region coordinates are evaluated. For example, if the initial coordinates of the screen are given as (0, 0,800,600); the region coordinates of 'upper-right' is evaluated as (800/2 = 400, 0, 800, 600/2 = 300).

A similar problem also occurs in interval queries. When the user enters the phrase *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. To evaluate the intervals, duration of the video should be entered at the beginning of the querying.

- When the *last x minutes* is parsed from the query:
  Starting time of the interval = Duration of the video – x
  Ending time of the interval   = Duration of the video
- When the *first x minutes* is parsed from the query:
  Starting time of the interval = 0th minute
  Ending time of the interval = 0 + x

The information extraction module always returns a result. This result will be matched to the video data model to get the exact outputs. But what if, the video data model can not find an exact match with the semantic representations. For example if the query asks for the object ‘cat’, the system will return no result from the video data model if it does not contain an object ‘cat’. A similar problem can occur if the user enters more generalized or more specific words in the query. For example, the user can ask for retrieving frames involving a Mercedes. What if, the data model involves only the object car or vice versa? The solution to these problems will be described in the next chapter.

CHAPTER 5

CONCEPTUAL ONTOLOGY

In this chapter, the ontology part used in the thesis will be explained in detail. After parsing the query and extracting the objects, the semantic representations and the video data are matched directly. Whenever an object is found in the query, it is expected to be involved also in the video objects. By this way, the query can be executed. But, what if the object is not in the video object list? No matched results will be returned to the user. The user identifies the objects with his judgment as more general or more specific. For example, there is a car in the frame and the word *car* is used at the data entry phase. That is, the object has been entered as 'car' to the video database. But the user might query by using the word 'vehicle' instead of 'car'. A conceptual ontology is decided to be used, not to return null results.

Ontology is a kind of knowledge base that involves concepts and their definitions to use for the semantics of the application domain [46]. There are different types of ontologies like WordNet [57], SENSUS, GALEN etc. WordNet is the most used ontology for semantic similarity of nouns and verbs. The most similar concepts are returned to the user by evaluating semantic similarity between objects in video data and query object using the ontology. The methods for evaluating semantic similarity and the comparisons of the methods are also described in this chapter.

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. Minimum set of the related concept is 'synonym sets' or 'synsets' [41]. 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 of 'person' has a synset of {person, individual, someone, somebody, mortal, human, soul}. All these words can represent the concept 'person'.

Some other relations between synsets also exist. Therefore, a network structure occurs in the model. The relations given in Table 5.1 are taken from [40].

Table 5.1 Relations defined in WordNet

Relation	Description	Example
Hypernym	is a generalization of	*furniture* is a hypernym of *chair*
Hyponym	is a kind of	*chair* is a hyponym of *furniture*
Troponym	is a way to	*amble* is a troponym of *walk*
Meronym	is part/substance/member of	*wheel* is a (part) meronym of a *bicycle*
Holonym	contains part	*Bicycle* is a holonym of a *wheel*
Antonym	opposite of	*ascend* is an antonym of *descend*
Attribute	attribute of	*Heavy* is an attribute of *weight*
Entailment	entails	*ploughing* entails *digging*
Cause	cause to	to *offend* causes to *resent*
Also see	related verb	to *lodge* is related to *reside*
Similar	to similar	to *kill* is similar to *assassinated*
Participle of	is participle of	*stored* (adj) is the participle of "to *store*"
Pertainym of	pertains to	*radial* pertains to *radius*

The WordNet has an IS-A hierarchy model that can be thought of a tree having one root. However, some of the nodes in WordNet can have more than one parent. Multiple inheritance is used at a percentage of 2.28 % among this taxonomy [12].

E.g. The word 'globe' has two parents: 'model' and 'sphere'.

For this reason, WordNet is not exactly a tree. Figure 5.1 shows an exception.

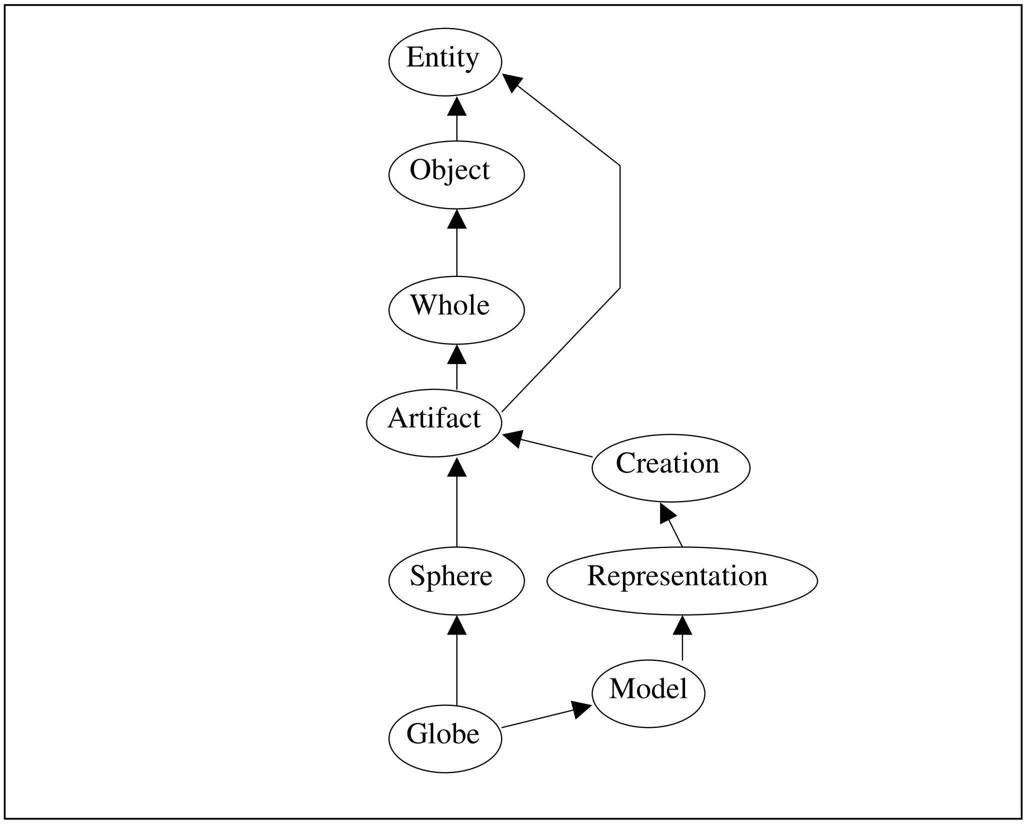


Figure 5.1 Example for multiple inheritance in IS-A hierarchy

In WordNet, there are nearly 75.000 concepts defined in tree-like structures where nodes are linked by relations. There are main hierarchies (or trees) for nouns. These are:

- Entity
- Abstraction
- Group
- Act, human action, human activity
- Psychological feature
- State
- Phenomenon
- Event
- Possession

In verbs, there are 628 separate hierarchies because of the fact that the relatedness between the concepts is less than that of nouns. Figure 5.2 taken from [40] shows a small subset of the entity hierarchy in the WordNet. In WordNet, not only nouns and verbs exist as concepts but also adjectives and adverbs occur.

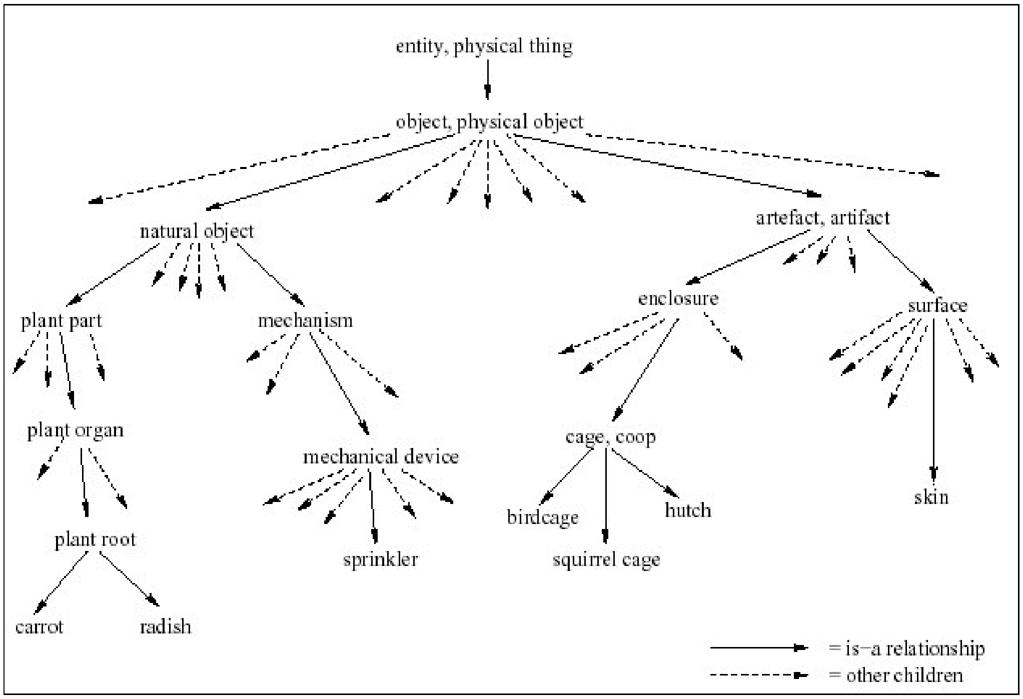


Figure 5.2 Entity hierarchy model in WordNet

There are two file types for each syntactic category. *Index files* include the word forms in lower cases and they are sorted in ASCII character set. This helps to make binary search over the files. Each line in the file contains the sense count, relational pointers to synsets, offsets of the entry in data file and the word forms

[54]. Second type of files is the *data files*. Data files contain the information about the synsets that are defined in index files.

WordNet also has an API that provides some functions to use it. These functions abstract the programmer from the complex table structures. They perform searches, retrievals and even morphological analysis of the words [54]. We used the version 2.0 of WordNet all through the thesis.

5.2 Ontology Based Querying

The aim is to use the knowledge of domain-independent semantic concepts to get better and closer results from the query. The main issue in semantic similarity is getting more accurate results between two concepts: the video object and the query word [3]. In the thesis, semantic similarity is used for the 'word', not for the 'concept', because 'concept' is a sense of a word. However no word sense disambiguation (WSD) method is used to find the sense of the query words and objects. The WSD problem is a common problem in natural language processing.

The factors that determine the semantic similarity are the IS-A hierarchy and synonym relation between concepts [3, 7]. IS-A relation concerns with the generalization and specification degrees. Instead of constructing a domain-based ontology, a domain-independent semantic ontology (WordNet) is decided to be used for conceptual similarity.

The main concern is how to benefit from ontology. Two approaches are considered as alternatives depending on the video data model structure and the previous studies over ontology and semantic similarity. These are the followings:

- Expanding the query word by using semantic relations and comparing it with objects in video data.
- Evaluating semantic similarity between query word and every object in video data.

5.2.1 Method of Expanding the Query Word

This approach depends on expanding the query word using hyponym (IS-A) / hypernym (HAS-A) and synonym until a cut-off limit. The result set will then be matched with the objects of the video database. The depth, the type of relation and the density of the level determine the similarity degrees of the elements in the result set. The video objects are matched one by one starting from the concept with the highest degree from the result set. The number of elements in the result set can be limited with a cut-off value after analyzing the performance measures. The maximum level for the expansion and the number of elements in the result set can also be the factors for the cut-off value.

For example, in the query, if the word *car* is entered, the concept will be expanded into words like:

Synonym : {auto, automobile, machine, motorcar, railcar, railway car, railroad car, cable car, gondola, elevator car}

Hypernym : {motor vehicle, automotive vehicle, wheeled vehicle, compartment} *(expanded for only one level)*

Hyponym : {ambulance, beach wagon, bus, cab, baggage car, cabin car, jeep etc.}

This method is especially used in Internet. In [48], keywords taken from the web document are extended by using the ontology and the results are used for creating the representation of a web document. For the reason that the vocabulary of the web ontology is insufficient, WordNet is used to enrich the ontology. By using synonym, hypernym/hyponym and meronym/holonym relations, the word with the given sense is expanded and linked to the web ontology. As another study in [55], information retrieval accuracy is improved by adding synonyms and direct related synsets that have relative weight (α) of 0.5 to the query vector.

Expansion of concepts is also used to query in web. For web queries, sense based query expansion is an approach that has not proved its effectiveness. This method relies on replacing senses with synonyms and hypernyms. Instead of replacing senses with the related concepts, Navigli and Velardi [36] have implemented an ontology-based query expansion method. Not only synonym and hypernym relations are used, but also some extra relations are implemented like *gloss*, *topic* and *domain* to expand the query. For each word in the query, the senses of the words are expanded by these relations to create semantic networks. And these networks are crossed to find common nodes. The number of common nodes expresses the degree of the accurate sense of the query.

However in this method it is difficult to determine the cut-off situations. Also the expansion is done for all senses of the query words, since their senses are not known. This is a very exhaustive operation. Also every element in the result set has to be matched with the objects in the video database. Determining an ineffective cut-off value causes an unnecessary result set with irrelevant elements . If the number of video objects is n_{VD}, and number of elements in the result set is $m_{RS,}$ the complexity of matching will be $O(m_{RS} * n_{VD})$.

5.2.2 Evaluating Semantic Similarity

By using another algorithm, many projects are evaluating semantic similarity, since more accurate results are obtained. The algorithms which will be described in Section 5.3 are mostly written by using WordNet ontology. Since our system also uses WordNet, it brings an advantage for the implementation. The semantic similarity can be evaluated between the query word and the objects of video data, instead of just focusing on the query word as in the previous method. Especially, the relations of hypernym and synonym are used in this technique.

The algorithm is based on finding the query word and an object of the video database in WordNet and evaluating the semantic similarity degree between the query word and the object for all sense pairs. The highest sense pair's similarity value is taken as the video object's similarity degree for the query word. This operation is done for every video object. Similarity values are sorted in descending order and, the resulting set of objects is returned according to a cut-off value.

Since the user does not enter any information about senses of the words, the senses of both the query word and the video object should be evaluated for semantic similarity. The complexity of this approach is less than the previous method, because only one word (query word) will be matched with the video objects. The complexity of the algorithm is $O(n_{VD})$ where n_{VD} is the number of objects in the video database.

In the thesis, this approach is preferred since it has a lower complexity and higher accuracy. Moreover, since we do not know the senses of query words, this method is more suitable than the previous one for our system.

5.3 Methods for Semantic Similarity

The issues in semantic similarity measuring are word sense disambiguation, determining the structure of texts, text summarization and annotation, information extraction and retrieval, automatic indexing, lexical selection and automatic correction of word errors in text [8]. The methods for conceptual similarity are divided into three groups:

- Distance Based Similarity
- Information Based Similarity
- Gloss Based Similarity

These methods will be summarized in the following subsections. Before explaining the methods, the notation and definitions should be given:

- len (c_1, c_2): It is the notation for shortest path, also meaning 'length' in WordNet from synset c_1 to the synset c_2.
- depth (c_1): It is the depth of the synset c_1. Depth means the length of the path to the root, therefore depth (c_1) = len (root, c_2).
- lso (c_1, c_2) : It represents the lowest super-ordinate (most specific common subsumer) of c_1 and c_2.

5.3.1 Distance Based Similarity Methods

A distance based similarity method is also called edge-based method because it depends on counting edges in a tree or graph based ontology. Finding the shortest path is important, because in WordNet there can be many ways between two synsets due to multiple inheritance. As explained before, there are nine hierarchies for nouns in WordNet. This means nine separate tree-like structures. There will be a gap when a path is tried to be found between the concepts in two different hierarchies. To solve this problem, Budanitsky and Hirst [8] propose to put an imaginary root on top of all hierarchies. So the length and depth are evaluated by considering this root.

The method is simple but it has some disadvantages. It is not so realistic to evaluate the length of the path between two nodes if the edges are not weighted. The reasons are:

- *The density of the network* is not constant [28]. The number of nodes per level increases as the level increases. Therefore at lower levels, the density is higher. More general concepts stay at the higher levels. This causes the length on higher levels having more general semantics than the lower levels which have denser density. For instance both plant/animal (higher levels) and zebra/horse (lower levels) has a distance of 2. But zebra/horse has a closer semantics than plant/animal.
- *The link type* is important for the metrics. IS-A hierarchy is the most common relation that is used for similarity. However, if other relations like meronym/holonym or antonym are also considered, the weight of edges will differ [20].
- *The relation among the siblings* is also a factor affecting the weights of edges. Not every child has the same closeness to its parent. Therefore a corpus should be used to get statistical value between siblings [20]. This method is studied by many researchers and the results have improved with the consideration of these factors.

The distance based similarity methods are:

- *Rada et al. (1989)'s Simple Edge Counting*

Rada and his colleagues used MeSH (Medical Subject Headings), a semantic network of 15000 terms rather than WordNet. BROADER-THAN relation is the link factor of edges between the nodes. The method is very simple. It just finds the shortest path (# of edges) between any two nodes in MeSH and pretends it as the conceptual distance [42].

Formula:

$$Rada = len(c_1, c_2)$$

In 1993, Lee also implemented the same method on WordNet by using IS-A hierarchy [40].

- *Leacock and Chodorow's Normalized Path Length*

By adding relative depth factor to path length, Leacock and Chodorow proposed a formula depending on the overall depth of the hierarchy tree using IS-A hierarchy.

Formula:

$$Leacock\ and\ Chodorow = \log (2D / len\ (c_1, c_2))$$

where *D* is the maximum depth of the taxonomy hierarchy. In WordNet 2.0, *D* is equal to 20.

To take the synonyms into the consideration, nodes are counted as the path length instead of edges. Therefore if two concepts are in the same synset, similarity will be one unit [7]. But it ignores the fact that similarity is higher at lower levels.

- *Wu and Palmer's Conceptual Similarity*

This method is also based on similarity with the factor of the path lengths by using IS-A hierarchy. The difference from Rada's is that it takes concept depths into consideration as well. The lowest common subsumer *lso* which is the most common specific concept is also evaluated.

Formula:

$$\mathit{Wu\ and\ Palmer} = \frac{2 * depth\ (lso(c_1, c_2))}{2 * depth(lso(c_1, c_2)) + \underbrace{len(c_1, lso(c_1, c_2)) + len(c_2, lso(c_1, c_2))}_{len\ (c_1, c_2)}}$$

Wu and Palmer [58] first studied this method on verbs in a hierarchy. The project separates different senses of verbs into different domains [7]. It is an interesting point that this formula can be effectively applied to any part of speech.

- *Hirst and St-Onge's Medium Strong Relations*

This measure differs from other methods by not depending on IS-A relation only. Therefore, this method can be used between heterogeneous pairs of parts of speech. This property is special to this method [17]. In this method, relatedness degrees are assigned to concepts. There are four types of relatedness:

- Extra strong: This property concerns with the surface form of the words. Therefore it is ignored for word sense evaluation.
- Strong : Two concepts are strong, if
  - They have instances of the same concept

- They associate with two different synsets that are connected by the antonyms (horizontal) relation.
- One of the concepts is represented by a compound word and other is represented by a word which is the part of a compound. There must be a synset relation between those two concepts.

- Medium-strong : If an allowable path exists between concepts
- Weak: If two concepts do not have any relatedness explained above

The relations defined in WordNet are represented with directions as *upward* (hypernymy and meronymy), *downward* (hyponymy and holonymy) and *horizontal* (antonym). The weight of the path depends on the length of the path and number of changes in direction.

Formula:

$$\textit{Hirst and St.Onge} = C - len\,(c_1, c_2) - (k * \textit{number-of-change-indirection between } c_1 \textit{ and } c_2)$$

where *C* and *k* are constants with values *C*=8 and *k*=1 from experiments.

- *Sussna's Depth-Relative Scaling*

Sussna's approach [50] is based on the fact that siblings at higher depths are closer than the ones at upper depths. The edges in the WordNet noun hierarchies are considered to be two directed edges representing inverse relations. Every relation of hypernym, hyponymy, holonymy and meronymy has weights and ranges between 1 and 2. Synonym relation gets a weight of 0, because it is a relation on a single node.

Formula :

$$\textit{weight of the relation types (r) leaving } c_1 = wt(c_1 \rightarrow r) = max_r - \frac{max_r - min_r}{edges_r(c_1)}$$

$edges_r(c_1)$: number of relations of type r leaving node c_1

$$\textit{Sussna's distance} = \frac{wt(c_1 \rightarrow r) + wt(c_2 \rightarrow r')}{2 * max\{depth(c_1), depth(c_2)\}}$$

where r is relation between c_1 and c_2 , r' is inverse of r.

This distance formula evaluates the strength between the source and target node [50].

Semantic similarity (c_A, c_B) = sum of Sussna's distance between pairs of adjacent nodes in the shortest path of c_A&c_B

This method focuses on the depth and density of the taxonomy.

- *Agirre and Rigau's Conceptual Density*

Agirre and Rigau [2] proposed this method especially for word sense disambiguation but they also formulized conceptual density, taking the region of the maximum density involving the senses of a concept [7]. The parameters in their formula depend:

- Shortest path length between the concepts
- Depth in the hierarchy (deeper concepts are related closer)
- Density of the concepts (concepts in higher density are related closer)

Formula :

$$Agirre\&Rigau's\ Conceptual\ Density\ (c, m) = \frac{\sum_{i=0}^{m-1} (n * hyp_c)^{i^{0.2}}}{descendants_c}$$

c: is the given concept at the top of the sub-hierarchy

m : number of senses in the sub-hierarchy c

descendants$_c$: number of concepts in the sub-hierarchy below c including c itself

$n * hyp_c$: mean of hyponyms per node in c's sub-hierarchy

5.3.2 Information Based Similarity

The methods for information based similarity use corpus in addition to the ontology in order to get statistical values. Implementing these methods is more difficult than evaluating path lengths.

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 *entity* has lower information content.

The information based similarity methods are:

- *Resnik's Information-based Combined Approach*

Information content is obtained by getting the frequency of the concept from a corpus and evaluating the probability.

$$Information\text{-}content\ of\ a\ concept = -\ log\ (p(c))$$

$$p(c)\ (probability\ of\ a\ concept) = \sum_{w \in W(c)} \frac{count(w)}{N}$$

where *W(c)* is the set of nouns in corpus, whose senses are subsumed by *c* and *N* is the number of nouns in the corpus.

As a result, concepts that are in the higher levels of the hierarchy will have higher counts in the corpus, higher probabilities, and directly lower information content. Semantic similarity of Resnik, uses IS-A hierarchy and depends on the logic that two concepts are proportionally related depending on the information degree they share. This is evaluated by taking the information content value of the lowest common subsumer [45].

Formula :

$$\boxed{Resnik = Information\text{-}content(\ lso(c_1,\ c_2\ ))}$$

The disadvantage of this formula is that many concepts can have the same least common subsumer and thus these can take the same similarity measure. For instance, in WordNet the concept of *vehicle* is the least common subsumer of *jumbo jet, tank, house trailer and ballistic missile*. Any pair of these will have the same semantic similarity [42].

- *Jiang and Conrath's Combined Approach*

Both edge counting and Resnik's method are combined in this method by using a corpus. Different from Resnik's, it takes the information content of not only lso, but also the concepts themselves.

Formula :

$$\boxed{\begin{aligned}&Jiang\ and\ Conrath\ distance\ (c_1,\ c_2) = inf.content(c_1) + inf.content(c_2) - \\ &\qquad\qquad 2 * inf.content(lso(c_1,\ c_2))\end{aligned}}$$

Concepts that are closer, semantically have a lower degree [20]. To maintain the true value of semantic similarity the formula is converted as :

$$\textit{Jiang and Conrath} = \frac{1}{dist_{J\&C}(c_{1,}\, c_2)}$$

- *Lin's Universal Similarity Approach*

This method which is based on 'Similarity Theorem' states that similarity is a relation of the common information content of two concepts to the information content of the concepts themselves. The more commonality they share, the more similar they are [30].

Formula :

$$Lin = \frac{2 * inf.content\ (lso\ (c_{1,}\, c_2))}{inf.content\ (c_1) + inf.content(c_2)}$$

5.3.3 Gloss Based Similarity

Gloss is the definition of a concept. For instance, *small and light boat pointed at both ends propelled with a paddle* is the gloss of *canoe*. By using the gloss of concepts, relatedness can be evaluated by overlapping the glosses. It has the advantage that similarity between different part of speech concepts can be compared. However, the gloss definitions are too short to be compared with other glosses. No overlap can exist if the glosses are not extended. Lesk's algorithm depends on WordNet to find the overlapping definitions of concepts and concepts to which they are related [42].

5.4 Comparison of Semantic Similarity Methods

In Section 5.3 semantic similarity methods are explained in detail. The question is which one is better. Researchers analyzed these methods over test data. The accuracy percentage is the most important criteria they considered, since 100% accuracy can not be obtained in NLP because of the word sense disambiguation problem and independence from rules.

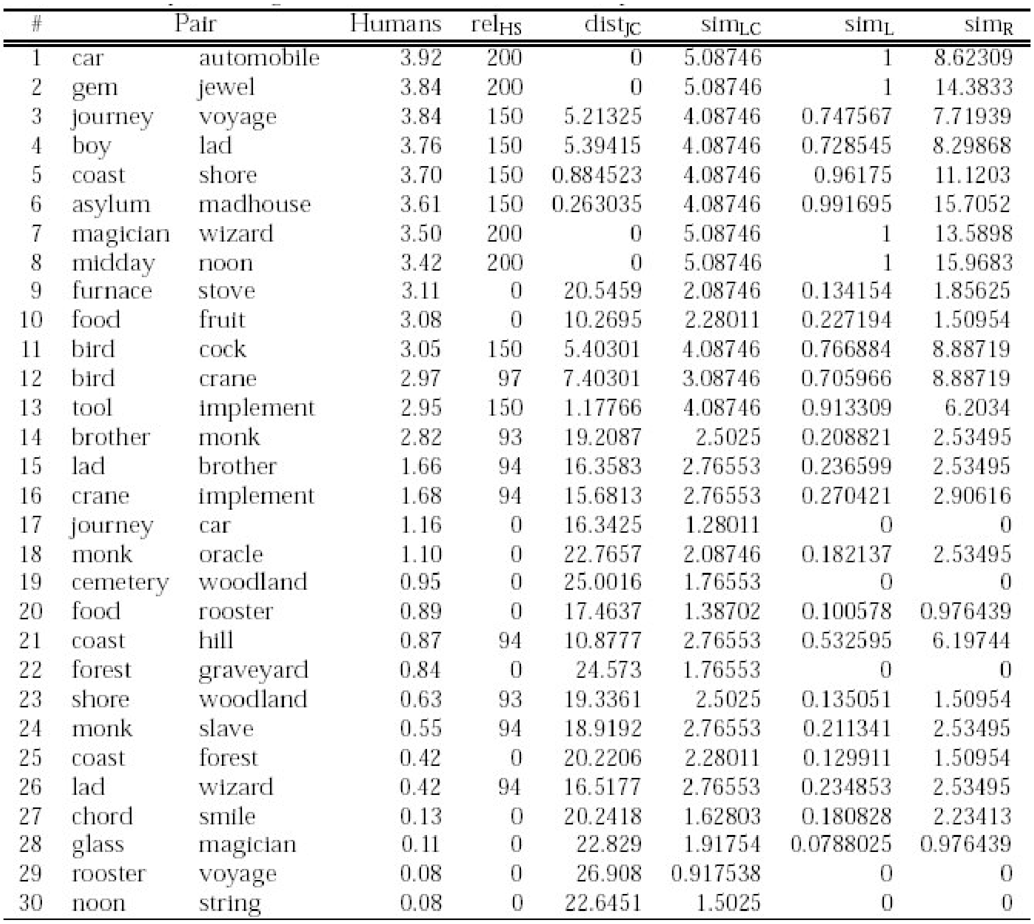
Miller&Charles [34] and Rubenstein&Goodenough [47] chose the best approach to compare the similarity methods as to observe if the human judgment is close to relatedness [8]. Test subjects are asked to give points between 1 and 4 to 65 pairs of words according to similarity of meanings. Five methods (i.e. Hirst-St &Onge, Jiang&Conrath, Leacock&Chodorow, Lin and Resnik's similarity results) are compared in their own scale with the human judgment points as in Table 5.2. For instance, the results obtained from the method of Resnik vary between 8.6 and 0, however results of Hirst-St&Onge method vary between 0 and 200. The higher values for results mean higher similarity obtained from the methods. In some cases 0 is obtained from the methods which show there is no similarity between two concepts. These situations explain if the methods can respond to any two concepts.

Another study on method comparison is experimented by Jiang and Conrath [20] comparing the methods with human judgment. Jiang and Conrath's method is a hybrid solution of edge-based and information content. Therefore, their experiment has the ability to show more general results over method classifications like node-based, edge-based and hybrid solutions (see Table 5.3). The correlations are evaluated for the solutions. The closer correlation value to the correlation of human judgment is pretended to be the better solution.

The experiments of Hirst-Budanitsky [9] and Patwardhan show that Adopted Lesk and Jiang&Conrath are more accurate methods than others. However the results are varying depending on the word pairs. For information content methods, the used corpus also affects the accuracy [41]. The methods depend on different sources. Leacock&Chodorow and Hirst&St-Onge relies on the concept hierarchies, however Resnik, Jiang&Conrath and Lin depend additionally on a corpus. Adopted

Lesk is different from others that it relies on the gloss and WordNet. It is seen from the results that there does not exist any best method. Only information content or only edge counting is not enough.

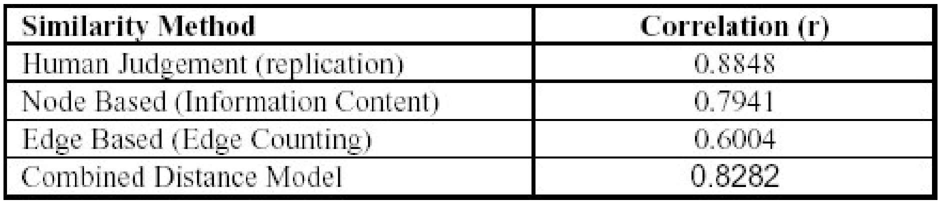
Table 5.2 Miller and Charles experiment



#	Pair		Humans	rel_{HS}	$dist_{JC}$	sim_{LC}	sim_L	sim_R
1	car	automobile	3.92	200	0	5.08746	1	8.62309
2	gem	jewel	3.84	200	0	5.08746	1	14.3833
3	journey	voyage	3.84	150	5.21325	4.08746	0.747567	7.71939
4	boy	lad	3.76	150	5.39415	4.08746	0.728545	8.29868
5	coast	shore	3.70	150	0.884523	4.08746	0.96175	11.1203
6	asylum	madhouse	3.61	150	0.263035	4.08746	0.991695	15.7052
7	magician	wizard	3.50	200	0	5.08746	1	13.5898
8	midday	noon	3.42	200	0	5.08746	1	15.9683
9	furnace	stove	3.11	0	20.5459	2.08746	0.134154	1.85625
10	food	fruit	3.08	0	10.2695	2.28011	0.227194	1.50954
11	bird	cock	3.05	150	5.40301	4.08746	0.766884	8.88719
12	bird	crane	2.97	97	7.40301	3.08746	0.705966	8.88719
13	tool	implement	2.95	150	1.17766	4.08746	0.913309	6.2034
14	brother	monk	2.82	93	19.2087	2.5025	0.208821	2.53495
15	lad	brother	1.66	94	16.3583	2.76553	0.236599	2.53495
16	crane	implement	1.68	94	15.6813	2.76553	0.270421	2.90616
17	journey	car	1.16	0	16.3425	1.28011	0	0
18	monk	oracle	1.10	0	22.7657	2.08746	0.182137	2.53495
19	cemetery	woodland	0.95	0	25.0016	1.76553	0	0
20	food	rooster	0.89	0	17.4637	1.38702	0.100578	0.976439
21	coast	hill	0.87	94	10.8777	2.76553	0.532595	6.19744
22	forest	graveyard	0.84	0	24.573	1.76553	0	0
23	shore	woodland	0.63	93	19.3361	2.5025	0.135051	1.50954
24	monk	slave	0.55	94	18.9192	2.76553	0.211341	2.53495
25	coast	forest	0.42	0	20.2206	2.28011	0.129911	1.50954
26	lad	wizard	0.42	94	16.5177	2.76553	0.234853	2.53495
27	chord	smile	0.13	0	20.2418	1.62803	0.180828	2.23413
28	glass	magician	0.11	0	22.829	1.91754	0.0788025	0.976439
29	rooster	voyage	0.08	0	26.908	0.917538	0	0
30	noon	string	0.08	0	22.6451	1.5025	0	0

The analysis in Table 5.3 shows that combined (Jiang and Conrath) model gives better results than information content and edge based methods. Also it shows that density and depth factor affect the results in edge counting [20].

Table 5.3 Experimental results of Jiang and Conrath



Similarity Method	Correlation (r)
Human Judgement (replication)	0.8848
Node Based (Information Content)	0.7941
Edge Based (Edge Counting)	0.6004
Combined Distance Model	0.8282

In [42], all methods are evaluated in the concept of F-measure which has the formula of:

F-measure = (2 * precision * recall) / (precision + recall)
precision = number of correct answers / number of answers given
recall = number of correct answers / number of instances

According to this evaluation, the highest score is taken by the extended gloss method and $dist_{JC}$. Among the edge based methods, sim_{WP} has the highest F-measure comparing to rel_{HS} and sim_{LC}.

5.4.1 Methods Used in Our System

Since ontology processing should have small amount of workload in our system, the fastest and relatively most accurate method is chosen. In order not to concern with the corpus and probability evaluation the methods of Wu & Palmer and Leacock & Chodorow are chosen to compare the accuracy of the results in our implementation. Another reason to prefer these methods is that they have easier implementations. However in our implementations, methods are extended to consider the effect of word sense disambiguation problem. A word sense disambiguation (WSD) algorithm is not implemented directly. However sense-based ambiguity is normalized by making some improvements in the search algorithm.

Three different search algorithms are implemented for both of the methods:

- The similarity is evaluated for all combinations of senses of two concepts (query word and video object).

- WordNet evaluates the frequencies of senses depending on a corpus and tags the senses. If these frequencies are used, information content method will also be used indirectly. Depending on these tags, only *tagged* senses of two concepts are taken into consideration during similarity evaluation.
- In WordNet, senses of concepts are ranked in the order of their usage in daily life. Therefore by considering these ranks, some 'weights' are given to the senses for the purpose of giving higher similarity scores to more occurred senses.

These three approaches for two methods (Wu&Palmer and Lea&Chodorow) are implemented to make a comparison. The advantage of these approaches is also to decrease the WSD effect on the system.

Another experiment over ontology processing is carried out to see if the concepts with known senses give more accurate results. To observe the results, video data objects are entered with their senses by the user. There can be wrong estimations, but previous experiments showed that human judgment still gives the most accurate results. Therefore it is used for comparison with the test data. The results are given in the Section 5.6. The experiments show that the method of Wu&Palmer gives the highest results when used with tagged sense approach.

5.5 Algorithm Used for the Ontology

The methods described above have been implemented for nouns. However the edge counting methods are not suitable for verbs since WordNet's verb hierarchy is shallow and not so structured. Only a small set of verb concepts actually occurs in the same hierarchy, since there are 628 verb hierarchies in WordNet. Since there are many verb hierarchies, no exact paths between verb concepts exist [42].

A recursive algorithm is used to find the shortest path between noun concepts. WordNet is storing hierarchies of concepts in a linked list structure. Each synset has links to the next sibling, to its parent and to its children. Every relation between nouns is represented by a number like 2 for hypernyms, 3 for hyponyms. In the algorithm only hypernym relation is used for the search.

The most important functions of WordNet are:

- *traceptrs_ds(SynsetPtr synptr, int ptr_type, int pos, int depth):* This function is used for recursive search which traces the path for synset pointed by *synptr* in the relation of *ptr_type. pos* indicates if it is noun/verb/adjective or adverb. To show that it will be used for nouns 1 will be assigned to *pos*. The parameter *depth* indicates whether the search is recursive or non-recursive. When it is used for hypernym, it returns the parent synset of the given synset.
- *findtheinfo_ds(char *searchstr, int pos, int ptr_type, int sense_num) :* It returns the synset structure for the given word of *searchstr*. *pos* and *ptr_type* are the same as above. *sense_num* is the sense number of the word synset. Before calling this method, WordNet's *morphword()* and *getindex()* functions are called. *morphword()* is a morphologic analyzer of WordNet. It returns the stem of the word. *getindex()* is used to return the index to the word synsets. When the index pointer is handled, the number of senses of the word can be found.

No similarity search is enabled for proper nouns, because only known proper nouns exist in WordNet like *America* and *George Bush*. Therefore exact matches are searched between the query concept and the video objects in proper nouns.

Verbs are activities in the video database. In order to find at least one match not only objects but also activities in the query are processed using ontology. If an activity is found in the query, any exact match or a synonym match is searched in the database. The stemming process is implemented over activities and query verbs. Then, the series of query verb synset are compared with the senses of all video activity synsets. Whenever a match is found, the synset of the video activity is added to the result set.

5.5.1 Similarity Evaluation

Two stacks, one for the video object and one for the query word are used to store their paths to the root. The synset of the query word is traced up to the root recursively on the hypernym relation. Each synset on the path is pushed to the 'query' stack. The function *traceptr_ds* is called to reach the parent synset, but for each level it is controlled if the synset has more than one parent. If so, other parent synset's path is also traced, because the aim is to find the shortest path. Therefore all the alternative paths must be searched.

After reaching the root of the query synset, video object synset are traced up to the root recursively. Their path is also stored in the 'video' stack in the same manner of query stack. The difference is that whenever a new synset is pushed to the video stack, it is controlled if this element is in the query stack. If a common synset is found, it is the least common subsumer, because the search algorithm starts from the nodes then goes upwards. At each step when going to upper levels, it is controlled if there is a common synset. Whenever it is found, searching process stops. And the elements in the stacks form the path between query and video words.

When the depth is evaluated for the formulas, an unreal root is thought over all noun hierarchies to find at least one path between any two concepts in different hierarchies. So the depth value in method formulas is increased by 1.

5.5.2 Nouns with All Senses

In this method, all senses of the two concepts are compared with each other. If the query word has *x* and video word has *y* senses; *x*y* combinations are evaluated and the path with the minimum length is taken. It is the least efficient and least accurate method, since the last senses of two words can have the shortest path length. However, as the number of senses increases, the frequency of the concept used in daily life decreases. The most unrelated senses of concepts can be combined in this method.

5.5.3 Nouns with Weighted Senses

When the number of senses is low, the frequency of sense increases. To resolve the problem described in the previous section, the senses are given weights depending on their row. The sense having the least number gets a higher weight. Similarity is inversely related to the length of the shortest path. So the weights can be arranged on the lengths of senses.

e.g. Suppose the 1st and 4th senses have the same path lengths. Since the 1st sense is weighted, its length will be decreased and it will have a higher similarity.

5.5.4 Nouns with Tagged Senses

The best accuracy is taken from this method. WordNet data structure evaluates the frequencies for each sense of the given concept from the corpus. It tags the senses if their frequency is higher than 0 and ranks these senses. An evaluation just on these senses increases both the performance and accuracy.

5.6 Comparison of Methods over Nouns

Two test data domains are formed with 40 words. One of them is taken from a video clip of a street camera. Other test data is taken from a TV serial. Some general estimated query words are selected, which are not among the video objects. Two methods (Wu&Palmer and Lea&Chodorow) including three approaches are compared by evaluating F-measure.

The similarity score range is [0, 1] for the method of Wu and Palmer. It is [0, 1.7] for Lea and Chodorow. Actually the highest value they can take are $\max_{WP}$ = 1.62 and $\max_{LC}$ = 1. It is planned to give 1.7 and 1 to the scores in the algorithm, if two words are in the same synset to give a priority to synsets.

There are two issues to discuss in the comparison of methods and analysis. One of them is to choose the method with the highest accuracy. Second is to give a cut-off value for the elements of result set. But this cut-off value varies depending on the preferred method. Combining two problems, different cut-off values are experimented over six methods using two test domains. For the TV serial video objects, words *person, furniture* and *girl* are used as the query words. Words *person* and *vehicle* are used for the street video objects. Precision and recall values are evaluated using the formulas given in Section 5.4.2. The accuracy rate is directly proportional with these two values. Whenever both of these values are high, a better accuracy will be obtained. Another measure to be considered is the F-measure which depends on both precision and recall. Getting the result of maximum 1 from F-measure means the best accuracy. The results of the experiments can be seen in Table 5.4.

Moreover, another experiment is implemented over the same video objects with their senses defined. The same domains and the query words are used. The cut-off value for the result set is observed controlling the average F-measure. The aim of this experiment was to show that if the senses were known, word sense disambiguation problem would be solved and more accurate results would be obtained. However to enable this opportunity, the user should input the sense numbers of the video objects while entering the data into the video database. Therefore, the user should be aware of the senses of the objects.

5.6.1 Results

The results show that tagged Wu&Palmer method gives the highest F-measure among the other methods with an average F-measure of 0.8. This method depends on the formula of Wu&Palmer by also considering the corpus of WordNet. The best accuracy is obtained when the cut-off value is 0.55 for non-sensed video objects, and 0.65 for sensed video objects. It is also seen that if the user enters the video objects with their senses, F-measure becomes 1. The highest precision and recall values of the methods are given for the varying cut-off values in Table 5.5.

As a result, Wu&Palmer's tagged sense method is preferred in our system.

Table 5.4 F-measures over methods with varying cut-off values

	Cut-off	**person$_1$**	**vehicle**	**person$_2$**	**furniture**	**girl**
Normal_WP	*≥0.65*	0,761905	0,833333	0,72	0,714286	0,571429
	≥ 0.7	0,761905	0,888889	0,72	0,833333	0,705882
	≥ 0.8	0,823529	0,571429	0,777778	0,833333	0,727273
Normal_LC	*≥ 1*	0,727273	0,526316	0,692308	0,714286	0,666667
	≥ 1.2	0,823529	0,571429	0,777778	0,727273	0,6
	≥ 1.3	0,823529	0,571429	0,777778	0,727273	0,6
Weighted_WP	*≥ 0.7*	0,875	0,5	0,875	0,5	0,714286
	≥ 0.8	0,875	0,571429	0,875	0,588235	0,714286
	≥0.85	0,933333	0,666667	0,875	0,625	0,615385
Weighted_LC	*≥ 1.3*	0,666667	0,4	0,695652	0,454545	0,4
	≥ 1.5	0,761905	0,470588	0,8	0,588235	0,555556
	≥ 1.6	0,761905	0,470588	0,8	0,588235	0,555556
Tagged_WP	*≥ 0.5*	1	0,666667	1	0,555556	0,8
	≥0.55	**1**	**0,666667**	**1**	**0,555556**	**0,8**
	≥0.6	1	0,666667	1	0,588235	0,714286
	≥0.7	1	0,5	0,941176	0,833333	0,666667
	≥0.8	0,933333	0,333333	0,8	0,833333	0,5
Tagged_LC	*≥0.9*	0,761905	0,444444	0,8	0,526316	0,666667
	≥ 1	1	0,8	0,941176	0,769231	0,666667
	≥ 1.1	0,933333	0,571429	0,8	0,833333	0,727273

Table 5.5 Tagged Wu&Palmer method's measures on varying cut-off values

		Average Precision	**Average Recall**	**Average f-measure**
Non-sensed	*≥ 0.5*	0,724542	0,96	0,804444
	≥0.55	**0,724542**	**0,96**	**0,804444**
	≥0.6	0,722619	0,926667	0,793838
	≥0.7	0,80952381	0,791111111	0,788235294
	≥0.8	0,942857	0,615	0,68
Sensed	*≥ 0.5*	0,727273	1	0,8125
	≥ 0.6	0,857143	1	0,916667
	≥ 0.65	**1**	**1**	**1**
	≥ 0.7	1	0,7	0,785714

CHAPTER 6

MAPPING NL INTERFACE TO VIDEO DATA MODEL

This chapter introduces the mapping of ontology results to the parameters of semantic representations of the queries. These representations are then matched to the query types defined in the video data model. The semantic representations are the results of the information extraction module as described in Chapter 4. They include words extracted from the query as parameters. When the ontology is applied to the implementation, more objects and more activities are obtained to be the parameters of the semantic representations. Therefore, the process of query mapping to semantic representations is expanded when ontology is added to the system.

6.1 Mapping Ontology Results to Semantic Representations

Whenever the query includes an object, an activity or an event, the ontology implementation is called for the object or the activity. *main_search* is the function to get similar video objects for the query objects and *find_verb* is to get similar activities. These objects and activities are stored in a result set. Since there may be more than one element in these result sets, for each element a semantic representation is built. Here is an example below:

Query: *Retrieve all frames where the cat is seen.*

Before the ontology implementation :

Semantic Representation: *RetrieveObj(ObjA) : frames.*
ObjA(cat,NULL,NULL).

After the ontology implementation :

Semantic Representation: *RetrieveObj(ObjA) : frames.*
RetrieveObj(ObjB) : frames.
ObjA(cat, NULL, NULL).
ObjA(dog, NULL, NULL).

The result set returned from the ontology implementation has two elements: *cat* and *dog*. *cat* and *dog* are the objects in the video database, since ontology implementation evaluates similarity over video objects. *cat* is retrieved since it is a direct match. *dog* is retrieved by the ontology as the most similar object to *cat*. For each element in the result set, semantic representation is formed.

Not only for objects but also for activities semantic representations are formed for each element in the result set returned from the ontology implementation.

The problem arises when the query includes an event with more than one object. Events include an activity and one or more objects. However, for all objects and the activity, ontology should be implemented. There is a function *multi_search* defined in the ontology implementation to call for events and multiple objects in the query. This function takes the objects and the activity as the parameters. The semantic similarity search is implemented for all parameters by calling *main_search* and *find_verb*. For each parameter, a result set is obtained. Result set elements are combined to get tuples to obtain only one result set. The elements in the result sets are ranked depending on their semantic similarity values. When these elements are combined to form tuples, the values of the elements are multiplied. The product of the combination represents the semantic similarity value of the tuple. The tuples are ranked depending on their similarity value in the final result set. For each tuple in the final result set, a semantic representation is formed similar to objects. An instance is given below:

Query: *Retrieve all frames where the cat is drinking the milk.*

Before the ontology implementation :

Semantic Representation: *RetrieveEvnt(EvntA) : frames.*
EvntA(ActA, ObjA, ObjB).
ActA(drink).
ObjA(cat, NULL, NULL).
ObjB(milk, NULL, NULL).

Let's abstract the representation as *RetrieveEvnt(drink, cat, milk) : frames* .

After the ontology implementation :

The similarity set for the query word

- *cat* is *{cat, dog}.*
- *milk* is *{water, food}.*
- *drink* is *{drink, imbibe}.*

The tuples in the result set will be : *{(drink, cat, water), (drink, cat, food), (drink, dog, water), (drink, dog, food), (imbibe, cat, water), (imbibe, cat, food), (imbibe, dog, water), (imbibe, dog, food) }*

Semantic Representation (abstracts): *RetrieveEvnt(drink, cat, milk) : frames.*
RetrieveEvnt(drink, cat, food) : frames.
RetrieveEvnt(drink, dog, water) : frames.
RetrieveEvnt(drink, dog, food) : frames.
RetrieveEvnt(imbibe, cat, water) : frames.
RetrieveEvnt(imbibe, cat, food) : frames.
RetrieveEvnt(imbibe, dog, water) : frames.
RetrieveEvnt(imbibe, dog, food) : frames.

6.2 Mapping Semantic Representations to the Video Database

Semantic representations are constructed for mapping with the video data model. In the implementation of the video data model, for each type of query there exists a function. These functions get the same parameters as the semantic representations. Since there is a semantic representation form for each query type, the representations can be mapped to the functions directly. The video data model and our natural language query interface implementations are independent systems.

Whenever a query is entered, our implementation is called from the video database. Our system must return the semantic representation of the query. However the name of the representation and the function of the query type are different. Therefore the video database must understand which type of query representation is mapped to the function. The problem is solved by assigning an id for each query type. In additional to the semantic representation, this id is also returned from our implementation to the video database. Since the parameters of the functions and the representations are the same, the appropriate function is called depending on the id. The parameters of the semantic representation are directly assigned to the parameters of the function.

As explained before, there can be more than one semantic representation because of the ontology. A temporal file is used to return these semantic representations to the video data model implementation. The id and parameters of the representations are written to this file line by line. An example is given below :

Query : *Retrieve all frames where the cat is drinking the milk.*

The format of the temporal file

```
Id: 3
drink, cat, milk
drink, cat, food
drink, dog, water
imbibe, cat, water
drink, dog, food
imbibe, cat, food
imbibe, dog, water
imbibe, dog, food
```

Since the id of the elementary event query is 3, 3 is set to the *id* field of the file. This query type has parameters as *activity, objectX, objectY.* Therefore tuples taken from semantic representations are written line by line.

The video database also knows the format of this file. Therefore for each line read from the file, the function of the query type is called. The output (frames, objects, intervals etc.) of each function call is then combined and returned to the user.

CHAPTER 7

CONCLUSION

The system described in this thesis uses a natural language interface to retrieve information from the video database. The video data model used in the thesis supports content-based spatio-temporal querying. To implement the natural language interface, 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 is used for mapping the semantic representation over the functions of the video data model. Conceptual ontology is implemented as a part of natural language interface. Using an ontological structure, WordNet, retrieves most similar objects or activities. 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 implementation to call appropriate query function.

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 that 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. Whenever the video data model is expanded to handle compound and conjunctive query types, the extraction rules will be expanded to handle more complex queries. Elliptical sentences will be handled by the system as a future study. This ability will add an extra advantage and flexibility to the system.

Ontology studies showed that if the user enters the video objects with their senses, the accuracy rate of the natural language processing would increase to approximately 100%. In the future study, we decide to implement a user-friendly interface that allows the user to choose the sense number of the video objects while entering the objects to the database.

Many different enterprises may benefit from the results of this thesis. For instance the security guards may benefit from the automatically querying of the security camera recordings. Journalist may benefit from querying the news archives easily. Or, digital film archives may contribute to researches in art.

REFERENCES

[1] Adali S, Candan KS, Chen S, Erol K, Subrahmanian VS, "The Advanced Video Information System: data structures and query processing", *Multimedia Systems, vol.4*, pp. 172-186, 1996.

[2] Agirre E, Rigau G, “Word Sense Disambiguation Using Conceptual Density”, *In Proceedings of the 16th International Conference on Computational Linguistics (COLING-96), pages 16{22, Copenhagen, Denmark,* 1996.

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

[4] 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.

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

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

[7] Budanitsky A, “Lexical Semantic Relatedness and Its Application in Natural Language Processing”, *Technical Report CSRG-390, Department of Computer Science, University of Toronto,* 1999.

[8] Budanitsky A, Hirst G, “Evaluating WordNet-based Measures of Lexical Semantic Relatedness”, *Submitted for publication*.

[9] Budanitsky A, Hirst G, “Semantic Distance in WordNet: An Experiment, Application-oriented Evaluation of Five Measures”, *Proc NAACL 2001 WordNet and Other Lexical Resources Workshop, 29-34, Pittsburgh,* 2001.

[10] Chang S, Chen W, Meng H, Sundaram H, Zhong D, "VideoQ: An automated content based video search system using visual cues", *Proceedings of ACM International Conference on Multimedia, Seattle, WA, November 9-13*, 1997.

[11] Decleir C, Hacid MS, Kouloumdjian J, "Modeling and Querying Video Databases", *Conference EUROMICRO, Multimedia and Communication Track, Vastras, Sweden*, pp. 492-498 , 1998.

[12] Devitt A, Vogel C, “The Topology of WordNet: Some Metrics”, *Petr Sojka, Karel Pala, Pavel Smrz, Christiane Fellbaum, Piek Vossen (Eds.): GWC 2004, Proceedings, pp. 106-111, Masaryk University, Brno,* 2003.

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

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

[15] Erozel G, Cicekli NK, Cicekli I, "Natural Language Interface on a Video data Model", *The IASTED International Conference on Databases and Applications DBA 2005, Innsbruck, Austria* , 2005.

[16] Hacid MC, Decleir C, Kouloumdjian, “A Database Approach for Modeling and Querying Video Data”, *IEEE Trans. On Knowledge and data Eng. 12(5), pp. 729-750,* 2000.

[17] Hirst G, St-Onge D, “Lexical Chains as Representationa of Context for the Detection and Correction of Malapropism”, *Christlane FeUbaum (ed.), MIT Press, Cambridge MA.,*1998.

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

[19] Informedia, Carnegie Mellon University, http://www.informedia.cs.cmu.edu/html/description.html, Last Updated on June 2005.

[20] Jiang JJ, Conrath DW, "Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy*", In Proceedings of International Conference on Research in Computational Linguistics (ROCLING X), Taiwan,* 1997.

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

[22] Knappe R, Bulskov H, Andreasen T, "Perspectives on Ontology-based Querying", *Heiner Stuckenschmidt (Eds.): 18th International Joint Conference on Artificial Intelligence, Ontologies and Distributed Systems, IJCAI 2003, Acapulco, Mexico, Workshop Program*, pp. 53-59, 2003.

[23] Koprulu M, "A Spatio-Temporal Video Data Model", *MSc Thesis, Middle East Technical University Graduate School of Natural and Applied Sciences, Department of Computer Engineering*, 2001.

[24] Koprulu M, Cicekli NK, Yazici A, "Spatio-temporal Querying in Video Databases", *Information Sciences 160, 2004, Elsevier Science*, pp. 131-152, 2004.

[25] Kuo TCT, Chen ALP, "Content-based Query Processing for Video Databases", *IEEE Transactions on Multimedia, 2(1):1-13,* 2000.

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

[27] Lee H, Smeaton AF, Furner J, "User Interface Issues for Browsing Digital Video", *21st BCS IRSG Colloquium on IR, Glasgow*, 1999.

[28] Lewis WD, "Measuring Conceptual Distance Using WordNet: The Design of a Metric for Measuring Semantic Similarity", *In R. Hayes, W. Lewis, E. Obryan, and T. Zamuner (Eds.), The University of Arizona Working Papers in Linguistics, Tucson*, 2002.

[29] Li J, Özsu M, Szafron D, Oria V, "Multimedia Extensions to Database Query Languages", *Tech. Report TR-97-01, Dept. Of Computing Science, The University of Alberta, Alberta, Canada,* 1997.

[30] Lin D, “An information-theoretic definition of Similarity”, *In Proceedings of International Conference on Machine Learning, Madison, Wisconsin,* 1998.

[31] LinkParser, http://bobo.link.cs.cmu.edu /link/, Last Updated on January 2005.

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

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

[34] Miller GA, Charles WG, “Contextual Correlates of Semantic Similarity”, *Language and Cognitive Processes, 6(1):1-28,* 1991.

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

[36] Navigli R, Velardi P, “An Analysis of Ontology-based Query Expansion Strategies”, *Workshop on Adaptive Text Extraction and Mining at the 14. European Conference on Machine Learning, Cavtat-Dubrovnik, Croatia,* 2003.

[37] Oomoto E, Tanaka K, "OVID: Design and Implementation of a Video Object database System", *IEEE Trans. On Knowledge and Data Engineering, 5(4)*, pp. 629-643, 1993.

[38] Palmer M, “Are WordNet Sense Distinctions Appropriate for Computational Lexicons?”, *Proceedings of Senseval, Siglex98, Brighton, England,* 1998.

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

[40] Patwardhan S, “Incorporating Dictionary and Corpus Information into a Context Vector Measure of Semantic Relatedness”, *MSc Thesis, the University of Minnesota Faculty of the Graduate School*, 2003.

[41] Patwardhan S, Banerjee S, Pedersen T, “Using measures of semantic relatedness for word sense disambiguation”, *in: Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics, Mexico City, pp. 241–257,* 2003.

[42] Pedersen T, Banerjee S, Pathwardan S, “Maximizing Semantic Relatedness to Perform Word Sense Disambiguation”, *University of Minnesota Supercomputing Institute, Research Report UMSI 2005/25, March,* 2005.

[43] Pradhan SRL, "A Query Model for Video Databases", *IEEE Transactions on Knowledge and Data Engineering October/November*, 2001.

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

[45] Resnik P, “Using Information Content to Evaluate Semantic Similarity in a Taxonomy”, *Journal of Artificial Intelligence Research 11,* 2001.

[46] Rodriguez MA, Egenhofer MJ, “Determining Semantic Similarity Among Entity Classes from Different Ontologies”, *IEEE Transactions on Knowledge and Data Engineering, 15, pp. 442-465,* 2003.

[47] Rubenstein H, Goodenough JB, “Contextual Correlates of Synonymy”, *Communications of the ACM, 8(10):627-633, October,* 1965.

[48] Sabrina T, Rosni A, T. Enyakong, “Extending Ontology Tree Using Nlp Technique”, *Proceedings of National Conference on Research & Development in Computer Science REDECS 2001, Selangor, Malaysia,* 2001.

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

[50] Sussna M, “Word Sense Disambiguation for Free-text Indexing Using a Massive Semantic Network”, *In Proceedings of the Second International Conference on Information and Knowledge Management, Arlington, Virginia,* 1993.

[51] Swanberg D, Shu CF, Jain R, “Knowledge guided parsing in video databases”, *Electronic Imaging: Science and Technology, San Jose, California,* 1993.

[52] Theodoridis Y, Vazirgiannis M, Sellis T, “Spatio-temporal indexing for large multimedia applications”, *Proc. IEEE Int. Conf. On Multimedia Computing Systems, Hiroshima, Japan,* 1996.

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

[54] Varma V, "Building Large Scale Ontology Networks", *Language Engineering Conference (LEC'02), Hyderabad, India, pp. 121- 127*, 2002.

[55] Voorhees EM, "On Expanding Query Vectors with Lexically Related Words", *Proceeding of the Second Text Retrieval Conference (TREC-2), NIST Special Publication, Gatherburg, Maryland,* 1993.

[56] Woods WA, Kaplan RM, Webber BN, "The Lunar Sciences Natural Language Information System: Final Report", *BBN Report 2378, Bolt Beranek and Newman Inc., Cambridge, Massachussetts,* 1972.

[57] WordNet 2.1, http://wordnet.princeton.edu/online/, Last Updated on 2005.

[58] Wu Z, Palmer M, "Verb Semantics and lexical selection", *In Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, pp. 133-138, Las Cruces, New Mexico,* 1994.

APPENDIX A

LINK TYPES USED FOR PARSING

O connects transitive verbs to their objects, direct or indirect

C links conjunctions to subjects of subordinate clauses

S connects subject nouns to finite verbs

P connects forms of the verb "be" to various words that can be its complements: prepositions, adjectives, and passive and progressive participles

G connects proper noun words together in series

MV connects verbs and adjectives to modifying phrases that follow, like adverbs, prepositional, subordinating, comparatives, participle phrases with commas, and other things.

J connects prepositions to their objects

M connects nouns to various kinds of post-noun modifiers: prepositional phrases, participle modifiers, prepositional relatives, and other kinds

A connects pre-noun ("attributive") adjectives to following nouns

AN connects noun-modifiers to following nouns

W connects the subjects of main clauses to the wall, in ordinary declaratives, imperatives, and most questions (except yes-no questions). It also connects coordinating conjunctions to following clauses

PF is used in certain questions with "be", when the complement need of "be" is satisfied by a preceding question word

Q is used in questions. It connects the wall to the auxiliary in simple yes-no questions; it connects the question word to the auxiliary in where-when-how questions

SI connects subject nouns to finite verbs in cases of subject-verb inversion

I connects infinitive verb forms to certain words such as modal verbs and "to"

DD connects definite determiners ("the", "his") to certain things like number expressions and adjectives acting as nouns

L connects certain determiners to superlative adjectives

APPENDIX B

THE RULES DEFINED FOR INFORMATION EXTRACTION

For return value:

- Control if there's an Op or Ce link. If so, the right-end will be the question-word.
- If Wq link is encountered, right-end of the link is the question word.

For object

- For questions like "in which frames is Monica seen?" PF/Q or Qd links are searched. Followed by an SI link's right-end is the subject or object of the sentence. If object is like "somebody, something, anything etc", it is an event.
- Control Cs link. The right-end of this link is an object.
- If there's a G link, then the proper nouns are formed of two or more names, so these words will be added to the object.
- Object rules involve in events, spatial relations etc. are described under those headings.

For activities:

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

- For questions like “in which frames is Monica seen?” PF/Q or Qd links are searched. Followed by an SI link's right-end is the subject or object of the sentence. If object is like “somebody, something, anything etc”, it is an activity.

For events:

- If the path is Ss+Pv, the right-end of Ss is the first object; the right-end of Pv is the activity. If MVp+Js follow Pv, the right-end of Js is the second object. If the path is Ss+Pg+Os; the right-end of Os is the second object and right-end of Pg is the activity.

For spatial relations:

- Pp+Js+Mp+Js (right-end of Ss is the first object, left-end of Mp is the relation, right-end of the Js is the second object). If at the right-end of Mp, there’s a right-end connector of an A/AN link, the left-end word of A/AN and right-end word of Mp forms the relation like right above, upper left etc. . If after Mp, there is an Mv link tracing, Mv ends forms the relation. “threshold”+Mp+Jp is for finding threshold (right-end of Jp is threshold value).
- Ss+Pp+Js(right-end of Ss is the first object, right-end of Pp is the relation, right-end of the Js is the second object), “threshold”+Mp+Jp is for finding threshold(right-end of Jp is threshold value)

For intervals:

- For questions like “on what interval” Wj link is searched. If found, a following Jp or Js is searched. Right end of Js/Jp is the question word.
- In interval questions, the intervals must be parsed. So a DD link is searched. Right-end of DD is start or end minutes of the interval. An L link is searched that has the same left-end of DD's. L's right-end is one of the

words of “last” or”first”. This word determines the boundaries of the interval.

For trajectories:

- After Ce (used for extracting question word), Mp+Js is traced. Js’s right-end word is the object that is trajected. Or Jr link is searched in the sentence. If it is found, right-end word of the link is the object of the query.
- Two MVp links are searched for the starting and ending trajectory regions. When they are found, the right-end words are processed as regions and regional rules are implemented for both of them.

For regions:

- Search for MVp link. The right word of the MVp link is the region keyword. If there is a Js link tracing and an AN link that has the same right-end connector with Js, the left-end word of AN is the region keyword. If there’s no AN link, an Mv link is traced after Js. If so, the left-end and right-end of Mv forms the region like upper-left, right-above etc. . . .

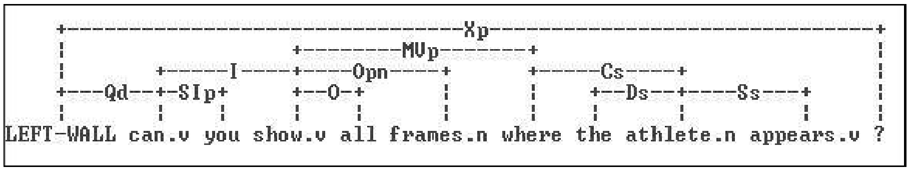
APPENDIX C

QUERY EXAMPLES

Elementary Object Query

- *Query* : Can you show all frames where the athlete appears?

  Output of the Parser :



```
    +--------------------------------Xp-------------------------------+
    |                  +--------MVp-------+                           |
    |       +-----I----+----Opn----+      +-----Cs----+               |
    +---Qd--+-SIp+     +--O-+      |      |    +--Ds--+----Ss---+     |
    |       |    |     |    |      |      |    |      |         |     |
LEFT-WALL can.v you show.v all frames.n where the athlete.n appears.v ?
```

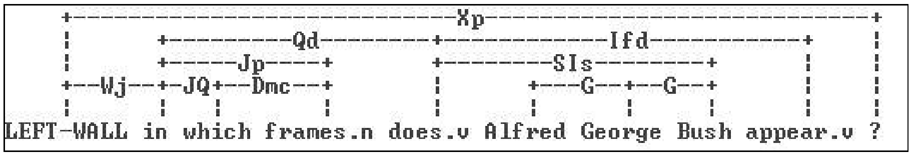
  Semantic Representation:

  RetrieveObj(Obj_A): frames.

  Obj_A (athlete, NULL, NULL).

- *Query* : In which frames does Alfred George Bush appear?

  Output of the Parser :



```
    +---------------------------Xp-----------------------------+
    |      +---------Qd--------+-----------Ifd------------+    |
    |      +-----Jp----+       +-------SIs---------+      |    |
    +--Wj--+-JQ+--Dmc--+       |      +--G---+--G--+      |    |
    |      |   |       |       |      |      |     |      |    |
LEFT-WALL in which frames.n does.v Alfred George Bush appear.v ?
```

Semantic Representation:

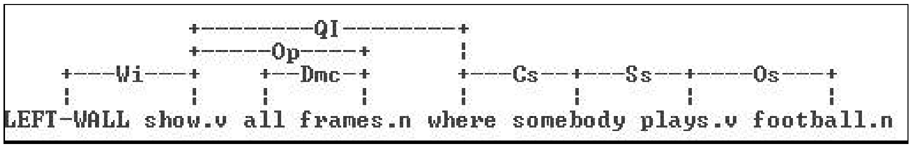
RetrieveObj(Obj_A): frames.

Obj_A (Alfred George Bush, NULL, NULL).

Elementary Activity Query

- *Query :* Show all frames where somebody plays football.

  Output of the Parser :



```
            +--------QI---------+
            +-----Op-----+      |
    +---Wi--+     +--Dmc-+      +--Cs--+---Ss---+---Os---+
    |       |     |      |      |      |        |        |
LEFT-WALL show.v all frames.n where somebody plays.v football.n
```

Semantic Representation:

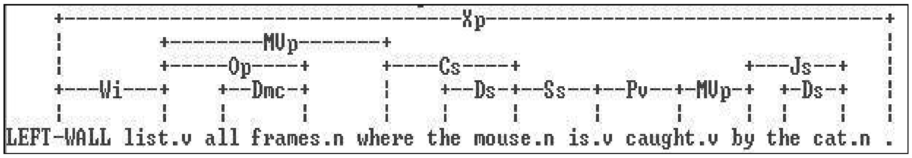
RetrieveAct(Act_A): frames.

Act_A (play football).

Elementary Event Query

- *Query :* List all frames where the mouse is caught by the cat.

  Output of the Parser :



```
    +---------------------------------Xp-----------------------------------+
    |        +--------MVp-------+                                          |
    |        +-----Op----+      +----Cs----+                   +---Js--+   |
    +---Wi---+    +--Dmc-+      |    +--Ds-+--Ss--+--Pv--+-MVp-+  +-Ds-+   |
    |        |    |      |      |    |     |      |      |     |  |    |   |
LEFT-WALL list.v all frames.n where the mouse.n is.v caught.v by the cat.n .
```

Semantic Representation:

RetrieveEvnt (Evt_A): frames.

Evt_A (Act_A, Obj_A, Obj_B).

Act_A (catch).

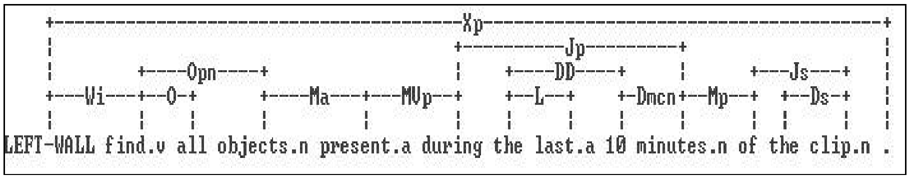
Obj_A (mouse, NULL, NULL).

Obj_B (cat, NULL, NULL).

Object Occurrence Query

- *Query :* Find all objects present during the last 10 minutes of the clip

  Output of the Parser :



```
    +----------------------------------------Xp---------------------------------------+
    |                                       +---------Jp----------+                   |
    |        +----Opn----+                  |    +----DD----+     |      +---Js---+   |
    +---Wi---+--O-+      +----Ma---+---MVp--+    +--L--+    +-Dmcn+--Mp--+  +--Ds-+   |
    |        |    |      |         |        |    |     |    |     |      |  |     |   |
LEFT-WALL find.v all objects.n present.a during the last.a 10 minutes.n of the clip.n .
```

Semantic Representation:

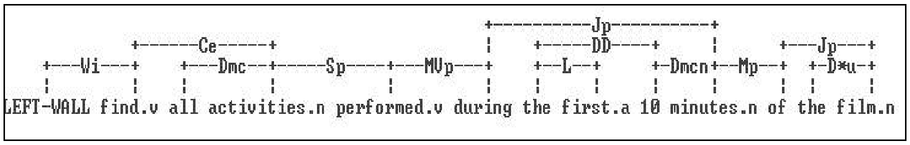
RetrieveIntObj (Int_A): objects.

Int_A (40, 50). *(assuming that the temporal length of the clip is 50 minutes)*

Activity Occurrence Query

- *Query :* Find all activities performed during the first 10 minutes of the film

  Output of the Parser :



```
                                                 +----------Jp----------+
             +------Ce-----+                     |    +-----DD----+     |      +---Jp---+
    +---Wi---+    +---Dmc--+-----Sp----+---MVp---+    +--L--+     +-Dmcn+--Mp--+  +-D*u-+
    |        |    |        |           |         |    |     |     |     |      |  |     |
LEFT-WALL find.v all activities.n performed.v during the first.a 10 minutes.n of the film.n
```

Semantic Representation:

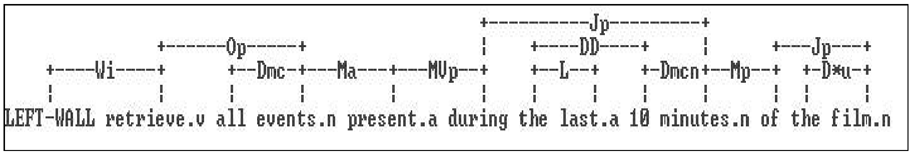
RetrieveIntAct_A (Int_A): activities.

Int_A (0, 10).

Event Occurrence Query

- *Query :* Retrieve all events present during the last 5 minutes of the clip

Output of the Parser :



```
                                               +----------Jp---------+
               +------Op-----+                 ¦    +----DD----+     ¦      +---Jp---+
    +----Wi----+      +--Dmc-+---Ma---+---MVp--+    +--L--+    +-Dmcn+--Mp--+  +-D*u-+
    ¦          ¦      ¦      ¦        ¦        ¦    ¦     ¦    ¦     ¦      ¦  ¦     ¦
LEFT-WALL retrieve.v all events.n present.a during the last.a 10 minutes.n of the film.n
```

Semantic Representation:

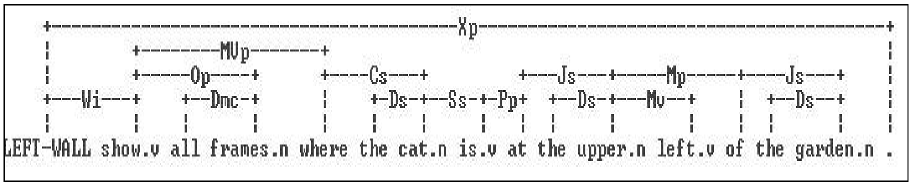
RetrieveIntEvt (Int_A): events.

Int_A (40, 50). *(assuming that the temporal length of the clip is 50 minutes)*

<u>Fuzzy Spatial Relationship Query</u>

- *Query :* Show all frames where the cat is at the upper left of the garden.

  Output of the Parser :



```
    +-----------------------------------------Xp-----------------------------------------+
    ¦        +--------MVp-------+                                                        ¦
    ¦        +-----Op----+      +----Cs---+         +---Js---+-----Mp-----+----Js---+    ¦
    +---Wi---+    +--Dmc-+      ¦    +-Ds-+--Ss-+-Pp+  +--Ds-+---Mv--+    ¦  +--Ds--+    ¦
    ¦        ¦    ¦      ¦      ¦    ¦    ¦     ¦   ¦  ¦     ¦       ¦    ¦  ¦      ¦    ¦
LEFT-WALL show.v all frames.n where the cat.n is.v at the upper.n left.v of the garden.n .
```

Semantic Representation:

RetrieveObj_ObjRel (UPPER_LEFT, 1): frames.

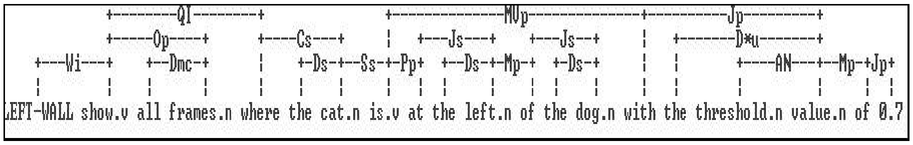
UPPER_LEFT (Obj_A, Obj_B).

Obj_A (cat, NULL, NULL).

Obj_B (garden, NULL, NULL).

- *Query :* Show all frames where the cat is at the left of the dog with the threshold value of 0.7

Output of the Parser :



```
             +--------QI--------+               +--------------MVp--------------+---------Jp----------+
             +-----Op----+      +----Cs---+     |   +---Js---+    +---Js--+     |    +-------D*u------+
    +---Wi---+    +--Dmc-+      |    +-Ds-+--Ss-+-Pp+  +--Ds-+-Mp-+  +-Ds-+     |    |      +----AN---+--Mp-+Jp+
    |        |    |      |      |    |    |     |   |  |     |    |  |    |     |    |      |         |     |  |
LEFT-WALL show.v all frames.n where the cat.n is.v at the left.n of the dog.n with the threshold.n value.n of 0.7
```

Semantic Representation:

RetrieveObj_ObjRel (LEFT, 0.7): frames.

LEFT (Obj_A, Obj_B).

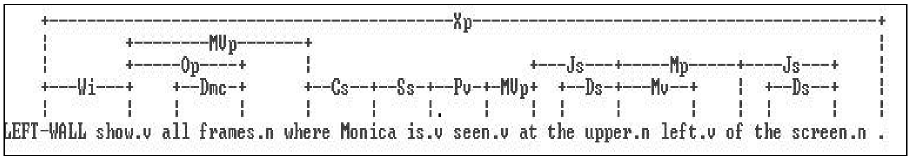
Obj_A (cat, NULL, NULL).

Obj_B (dog, NULL, NULL).

Regional (Frame) Query

- *Query :* Show all frames where Monica is seen at the upper left of the screen.

Output of the Parser :



```
    +-------------------------------------------Xp-------------------------------------------+
    |        +--------MVp-------+                                                            |
    |        +-----Op----+      |                       +---Js---+-----Mp-----+----Js---+    |
    +---Wi---+    +--Dmc-+      +--Cs--+--Ss-+--Pv-+-MVp+  +--Ds-+---Mv--+    |  +--Ds--+    |
    |        |    |      |      |      |     |.    |    |  |     |       |    |  |      |    |
LEFT-WALL show.v all frames.n where Monica is.v seen.v at the upper.n left.v of the screen.n .
```

Semantic Representation:

RetrieveObjReg (Obj_A, Reg_A): frames.

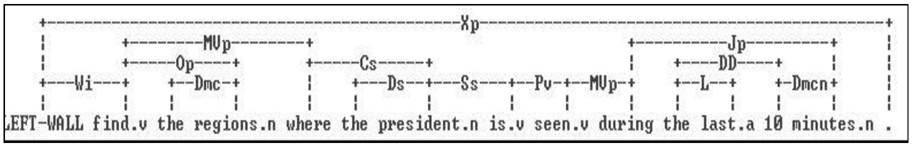
Obj_A (Monica).

Reg_A(0, 0, 50, 0). [*If coordinates of the frame's rectangle is considered as 0,0,100,100*]

Regional (Interval) Query

- *Query :* Find the regions where the girl is seen during the last 10 minutes.

Output of the Parser :



```
    +--------------------------------------------Xp---------------------------------------------+
    |        +--------MVp--------+                                  +----------Jp---------+     |
    |        +-----Op----+       +-----Cs-----+                     |    +----DD----+     |     |
    +---Wi---+    +--Dmc-+       |    +---Ds--+---Ss---+--Pv-+--MVp-+    +--L--+    +-Dmcn+     |
    |        |    |      |       |    |       |        |     |      |    |     |    |     |     |
LEFT-WALL find.v the regions.n where the president.n is.v seen.v during the last.a 10 minutes.n .
```

Semantic Representation:

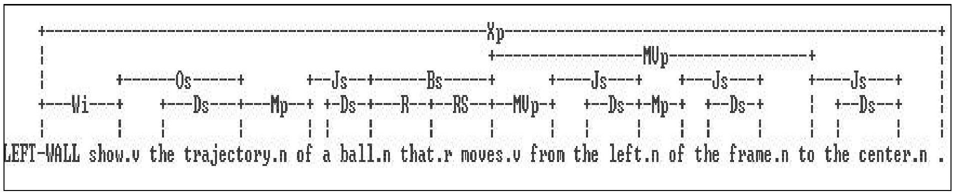
RetrieveObjInt (Obj_A, Int_A): regions.

Obj_A (president, NULL, NULL).

Int_A(40, 50). *(assuming that the temporal length of the clip is 50 minutes)*

<u>Trajectory Query</u>

- *Query* : Show the trajectory of a ball that moves from the left of the frame to the center.
- *Output of the Parser :*



```
    +---------------------------------------------------Xp--------------------------------------------------+
    |                                                   +-----------------MVp----------------+              |
    |        +------Os-----+       +--Js--+------Bs-----+      +----Js---+    +---Js---+     +----Js---+    |
    +---Wi---+    +---Ds---+---Mp--+ +-Ds-+---R--+--RS--+--MVp-+   +--Ds-+-Mp-+  +--Ds-+     |  +--Ds--+    |
    |        |    |        |       | |    |      |      |      |   |     |    |  |     |     |  |      |    |
LEFT-WALL show.v the trajectory.n of a ball.n that.r moves.v from the left.n of the frame.n to the center.n .
```

Semantic Representation:

TrajectoryReg (Obj_A, Reg_A, Reg_B): frames.

Obj_A (ball, NULL, NULL).

Reg_A (0, 0, 50, 100).

Reg_B(25, 25, 75, 75). [*If coordinates of the frame's rectangle is considered as 0,0,100,100*]