ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

COLLEGE OF NATURAL SCIENCES

DEPARTMENT OF COMPUTER SCIENCE

Natural Language Based Query Formulation for Video Retrieval Using Spatio-temporal Operators

Endris Osman Ali

A THESIS SUBMITTED TO

THE SCHOOL OF GRADUATE STUDIES OF ADDIS ABABA UNIVERSITY IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF MASTER OF SCIENCE IN COMPUTER SCIENCE

Addis Ababa, Ethiopia

January 2017

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

COLLEGE OF NATURAL SCIENCES

DEPARTMENT OF COMPUTER SCIENCE

Natural Language Based Query Formulation for Video Retrieval Using Spatio-temporal Query

Endris Osman Ali

Advisor: Fekade Getahun (PhD)

Signature of the Board of Examiners for Approval:

Name	**Signature**
1. Fekade Getahun (PhD)	_____
2. Solomon Atnafu (PhD)	_____
3. Yaregal Assabie (PhD)	_____

January 2017

DEDICATION

To My mother

Acknowledgements

Alhamdulillah, Thanks to almighty Allah the Creator and the Guardian for giving me the strength to finish this work. I offer my sincere gratitude to my advisor, Dr. Fekade Getahun, who has supported me throughout my thesis with his patience, motivation. His guidance and encouragement helped me through the learning process of the thesis work. Moreover, his advice in those challenging times put me on the right track.

A special thanks to my family. Words cannot express how grateful I am to them for all the sacrifices they've made on my behalf. Their trust and vote of confidence has kept me going thus far.

It is a great pleasure to acknowledge the efforts of my colleagues whose cooperation, friendship, and understanding were crucial to the completion of this thesis.

Table of Contents

List of Figures

List of Tables

List of Algorithms

List of Abbreviations and Acronyms

AVIS	Advanced Information System
MBR	Minimum Bounding Rectangle
MS	Microsoft
NL	Natural Language
NLIDB	Natural Language Interface in Database
NLP	Natural Language Processing
NLQ	Natural Language Query
OR-DBMS	Object Relational-Database Management System
POS	Part of Speech Tag
RDBMS	Relational Database Management System
SpatialOP	Spatial Operator
SQL	Structured Query Language
SRL	Semantic Role Labeling
SVO	Subject Verb and Object
TemporalOP	Temporal Operator
XML	Extended Markup Language

Abstract

High speed processors and large storage devices have made videos popular and easily available to everyone. With this dynamic growth, different research areas are created with content based video indexing, grouping, searching and retrieval. Video is processed based on the content of visual features and annotated metadata which is required for the identification of features, events and objects in a video.

One of the main problems of retrieving information from database is that casual users may not be aware of the structure of the database. The idea of using natural language, instead of SQL has prompted the development of query processing to fill this gap. Using natural language casual users can retrieve information on video database without the help of experts. The formulated queries can represent the stated object and event of user's query by using different spatial, temporal and predicate operators. So it would be easy to map and query relevant video scenes while preserving user stated objects and operators.

In this study we proposed natural language based query formulation for video retrieval at video scene level. Natural language queries are preprocessed to have ready for query formulation process. The query formulation process creates a high level SQL form which holds all necessary information to be ready for query execution phase. The designed data model supports a spatio-temporal and predicate based query for complex user queries. In addition, if the stated object query is not found in database user can retrieve semantically similar objects using the similarity metrics algorithm. A prototype application has been developed using appropriate tools and techniques for the soccer video domain. The prototype application has been tested after seen the full video content and checked with different query types. The system has been found to be 78% accurate to return video scenes that match with the user queries.

Keywords: Natural Language Processing, Natural Language Interface for Databases (NLIDB), Query Formulation, SQL, Similarity Metrics, Video Retrieval

Chapter 1: Introduction

1.1 Background

Video is a combination of multimedia objects including text, image, graphics and sound; and it has rich temporal and spatial relationships between its content objects; due to this it conveys large amounts of information. Video is a layered combination of frame, shot, scene and video. Among those video scenes provides a complete information or history unit. Many video techniques have been proposed regarding to handle the indexing, storage and retrieval process, from that video retrieval still has many research problem areas. Video retrieval is one of the most vibrant and important areas for both research and commercial applications. Designing effective video retrieval systems helps for applications such as digital libraries, video production and a variety of internet applications. This needs a great development and effective techniques for video retrieval and query formulation. Similarity-based video retrieval modeling and processing is described in [5] the authors used metadata information for representation of frames in a video and their approach allows spatio-temporal and content based retrieval. The presented approach is capable in responding to simple query such as "Search and play back the video frame containing the great goal of Dieago Armando Maradona".

Natural language querying (NLQ) for video database is described in [6]. Their work focuses on querying a natural language questions and retrieve results from the video repository. From this work, it is possible to answer question related to semantic, spatio-temporal relation from their video database.

Video retrieval is very dependent on querying the video structure or querying the content. The video structure has detailed description on segmentation and temporality of the video. Content-based video retrieval mainly focus on the raw video data or visual features such as color histogram, textures, shapes, or the semantic level that is the most complex. The concept formation approach maps visual features into higher level features [14].

Query formulation is a process of converting a written natural language text input into a structured format for better understanding by machines. Researches in query formulation are constantly evolving in big data text and with restricted organizational data set [52].

Query formulation is usually done into two main processing stages. The first processing stage is query refinement which mainly focuses on morphological structure and the second is query processing which translate the retrieval criteria using the specified language.

Morphological analyzer deals with the smallest part of the word which gives a predefined meaning by itself. In natural language processing analysis of each word and change those words into morphological level hires the query formulation process, because it helps to extract and analyze information like Part-of-Speech (POS), verb phrase and noun-phrases [53].

The approach in [5 and 6] proposed that video objects are annotated at frame level and user query involves events and has to be simple. As the user requirement on scene and it is complex the approach is not capable to answer user query.

1.2 Motivation

There are many natural language based video retrieval systems developed from earlier to now a day. Those proposed systems still require that the user is familiar with the queried knowledge structure. However, casual users need to be able to access the data despite their queries not matching exactly the queried data structures. In soccer video different objects and events interacts having different spatio-temporal activities. Querying video to data objects and events helps for sport analysts to easily search and to do further processing. Natural language query (NLQ) interface, gives a query solution which has exact match and similar concept from the video repository. In order to demonstrate our motivation, let us consider the following scenario a user wants to get a video scene in which "Messi and Fabregass appear together inside penalty area and Messi appear to the left of ball" from collection of video database. This query is complex written in natural language format, it involves Messi (player) and Fabregass (player), the event is inside, and Messi (player) and ball (object), the event is left operator. In order to retrieve such query, the query needs to be represented in standardized format. Thus, motivated by the drawbacks of existing natural language based video query systems. In this thesis user query terms are detected and refined for a better query formulation format. The formulated query handles object stated in the user queries as well as relationships between objects in a video. Moreover, a high level language query is generated associated to each user query. The video repository constitutes all video hierarchal models from object to video level and it works best at scene level because video scene gives complete information for the user query.

1.3 Statement of the Problem

Natural language based query formulation for video retrieval has been identified as means to formula in convenient way to get relevant videos from large multimedia database repository. NL query processing involves parsing the input statement, extract clauses, mapping clauses to structured query format, query the video and finally display the result in appropriate graphical format. On the other hand, researches mainly focuses on retrieving either simple text query or content based video retrieval and display the retrieved result in the form of video frame level or the video itself [3,5,6, 8].

NL query processing minimizes the overhead of query construction for non-computer expert users since to access data from the database it needs programming skill.

These results indicate that the need for natural language based video retrieval mainly support flexibility in searching, analyzing and extracting high-level complex query formulation and retrieving interest of user data based on similarity metrics. These are the most important challenges that need to be tackled while developing the research work.

This study tries to answer the following questions:

- How can standard query format is generated from complex NL query input?
- How a video scene is retrieved from large collection of video database?
- How the video data model is designed in order to access complex query?

In summary, this study focused on developing a video data model and implementation which accept a complex Natural Language (NL) query and translates into a structured query language form for retrieving relevant video scene. To evaluate the performance of these research works, we conduct a number of experiments and analysis from a collection of soccer video database.

1.4 Objectives

1.4.1 General Objective

The general objective of this research work is to develop and implement a natural language based query formulation and video retrieval model from video database.

1.4.2 Specific Objectives

The following are the specific objectives:

- ✓ Investigate the existing proposed video retrieval techniques
- ✓ Investigate the existing query formulation techniques
- ✓ Design a model for video database
- ✓ Design a model for natural language query formulation for video retrieval.
- ✓ Generate equivalent high level language query form
- ✓ Develop/adopt the necessary algorithms and implementation for natural language query formulation for video retrieval.
- ✓ Develop a prototype to demonstrate the practicability of the proposed model and implementation, and
- ✓ Evaluate the performance of the developed systems

1.5 Scope and Limitation

The main focus of this research work is to apply NL query text for retrieving video scene in large video database. The proposed system converts user queries into a standard query format (query formulation) on complex natural language questions.

This thesis work does not cover the annotation and query optimization issues in the process of video retrieval.

1.6 Methodology

Literature Review

We review different literature papers concerning natural language video retrieval and query formulation in order to analyze current trend works, techniques and approach. Different scholars have addressed this issue on different literature papers independently. We review these papers to identify the gaps and fill our contribution for further research.

Data Collection and Analysis

We will collect different video data samples of annotated videos from different multimedia sources over the web. The collected video samples are stored in repository system. The videos are processed to obtain the coordinate points of objects which are found in the video frames.

Design and Development

Different free and open source tools will be used during natural language query processing, query execution and taking coordinate points of objects. Java programming language will be used to implement the proposed solution while visual C+ with opencv programing language is used for dataset preparation. NetBeans IDE8.1 with Java programming language will be used to develop the prototype.

Evaluation

Finally we will evaluate the proposed design and implementation with different tasting strategies for checking the validity of the result. Proper testing will be made and the newly proposed solution will be evaluated in terms of its goals and contributions. A questionnaire will be prepared and distributed to different users and their rating will be used to evaluate the system. Accuracy of components will be evaluated using standard content based video processing metrics evaluation.

1.7 Application of Results

The outcome of this result can be used in different problem areas. Since the result has general nature, it can easily be adapted for any end users who are interested to collect videos on different issues for decision making purpose. Many applications are storing and retrieving information from databases to retrieve information and, this requires knowledge of database languages such as SQL. With the help of natural language query processing the task become easier for casual end users. When it comes to scene level video query, a video can be considered as a hierarchal structural form where a video scene is a table and the annotation is record at object level so that a video query language can be formulated and passed to the video to look for a specific scene. This can be applicable especially in news agencies where there are huge amount of news videos and retrieving one specific scene segment can be possible with a proper scene annotation dataset.

1.8 Thesis Outline

The rest of this thesis report is organized as follows. Chapter Two gives detail review on natural language query processing techniques, entity identification and similarity metrics methods. Chapter Three presents the review of related works. Chapter Four presents the proposed design and model of natural language based video query. Case studies in video scene retrieval for exact and similar query techniques for soccer video are presented in Chapter Five. In Chapter Six the

implementation of the prototype for the research will be presented. Moreover, tools and techniques that have been used for developing the prototype are stated and each component is implemented clearly along with user interfaces for demonstration. Finally, Chapter Seven concludes the overall work of this research work and, it presents contributions of the study and draws future directions.

Chapter 2: Literature Review

2.1 Introduction

High speed processors and large storage devices have made videos popular and easily available to everyone. Video conferencing, online videos, advertisements, sports, movies and news have gained popularity in today's technological advancement. From this perspective query the content of video is highly important for different user groups. There are different query techniques applied to retrieve the content of video database. The review discusses about natural language query techniques for annotated video database and the basic types of user query refinement to retrieve user query. Content based video retrieval systems are limited to use low-level features such as color, texture, shape, motion etc., and to formulate semantic level queries we need to have a metadata associated with the video content. Having this, query from the stored data requires knowledge in database languages and, users have to know the exact schema database, the roles of various entities, and the precise order to be followed. Now a day user query processing is shift towards to non-experts in database, so that designing user friendly query interface will have a more important factor for development in natural language interface over video database.

2.2 Video Structure and Content Representation

Videos are structured in hierarchy level forms which are objects, frames, shots and scenes on the temporal level [29]. A video object is defined as a collection of video regions that have similar roles under some criteria across several frames. Namely, a video object is a collection of regions exhibiting consistency across several frames in at least one feature. For example, "person walking" the person is the 'object' and walking would be segmented into a collection of adjoining regions. This region can be queried with features such as shape, color and texture. But all the regions may exhibit consistency in their motion attribute [32]. Rowe at el. [35] proposed indexing user access to large video database. The textual content is added to video content in different ways such as bibliographic, content dependent metadata and semantic metadata. Bibliographic metadata describes general video attributes, such as video title, its genre, owner, duration, and number of views on a video portal, but it does not describe directly the video content. Structural metadata for videos includes the hierarchy of scenes and shots. Content

dependent metadata annotate low level and intermediate level features like shape, color, texture, edge, motion, audio etc. Semantic metadata gives textual representation for the content of the image or video itself. The annotation text represents for entities like objects, events, motion and meaning of scenes [36]. Ideally, such metadata annotation should be accurate, complete and cost-effective to generate effective query result. Metadata generation for professional users are often quite extensive compared to user-generated content, since professionals includes detail information about the story line from the video content perspective. In contrast, most user-generated videos are only annotated with title, a short description, with some tags, and user ratings. Dessalegn Mequanint in [5] describes video as a content rich, characterized by voluminous and unstructured format and it conveys large amount of information. To manage video data effectively, the huge amount of video data must be structured into constituent parts of the video unit such as scenes, shots and key frames. Videos are created by taking a set of shots and composing them together using specified composition operators. A shot is defined as an image sequence that presents continuous action which is captured from a single operation of camera. Shots can be effectively considered as the smallest indexing unit where no changes in scene content can be perceived and higher level concepts are often constructed [18]. Meanwhile, video is segmented into shots and key frames to select representative story units. Video scene is a collection of semantically related and temporally adjacent shots which gives high level story units. Since, scenes are a composition of similar shots, the story units are constructed on the basis of key frames. For example, a person walking down a hallway into a room is one scene, even though there might be different camera angles shown. Three camera shots showing three different people walking down a hallway might be one scene if the important object was the hallway and not the people. A collection of scenes that represents a separable component of a movie is defined as a segment [31]. Generally a video stream contains several scenes or story units and each scene have a sequence of shots and frames respectively [5].

2.3 Semantic Annotation and Video Retrieval

Video annotation and retrieval is a hot research area due to huge amount of video production which enables to log most relevant description and other depth information to shots/scene in video stream. Many multimedia data objects are modeled with the use of text to describe the attributes information. Logging annotated video contents has two approaches, particularly production logging, which is carried out live or shortly after the event happened. The second one

is posterity logging which video material is logged offline and annotation is carried out using audio and captions [28]. Annotating video content with metadata has much more benefit such as we can extract and add information to the video content with less effort, and this helps to do certain operations on video browsing, search, analysis, retrieval, comparison, and categorization [36]. Having these roles, it also difficult giving correct representation for the right video content. Annotating video manually is time-consuming and inaccurate. Extracting semantic contents of textual tags in [30] are automatically extracted using low level image features like color shape, texture and motion. Semantic content of video can be queried in many different ways. Some example queries can be given as follows: For example for contained objects and events query like 'Retrieve all frames where the teacher is standing' and for spatial properties like 'Retrieve all frames where the ball is near the bar'. Spatial features are features that concern the geometric and topological aspects of salient video objects such as shape and position. The spatial feature is used to identify relations between video objects such as directional (right, left, above, front, etc.) and topological (touch, disjoint, overlap, equal, etc) relations. For example, in soccer video the spatial relationship between video objects can be used to answer queries like 'Give me all the videos in which Thery Henry appears to the left of Roberto Pirese'. Temporal features describe the temporal properties of the video object. The temporal feature is used to describe the temporal relationships among video objects such as (before, after, meets, overlaps, starts, during, finishes etc.). For example, in soccer video the temporal relationship between video objects can be used to answer queries like "Give me the videos in which 'Thery Henry' appears before 'Roberto Pirese'". Spatio-temporal operators define a mathematical relationship among salient video objects. For example in soccer video, the spatio-temporal relationships can be used to answer queries like 'Give me all the videos in which Thery Henry appears on the left of Roberto Pirese before Patric Viera'. For audio data query example is "Retrieve the audio data where Petra Berger talks at the concert video". Mostly video shots are tagged with text to retrieve semantic contents. However, extracting semantic content from the annotated video is limited to describe full activity of the event. Shot level video detection and annotation are widely used to retrieve video contents [40].

Video can be queried in different ways such as online video portals, using video search engines and research prototypes respectively [29]. These systems are tried to resolve the optimization search queries by minimize undesired media streaming initiations. Video portals store video contents on the servers and enable users to upload, annotate and judge those videos. Video

portals are designed to be search with text like YouTube [43]. Video search engines returns videos from several different video sources on the web, but do not store or provide videos by their own. Video search engines perform similar operation as usual web documents like MetaCrawler search engine [44]. The third technique is research prototypes, these systems have different scope than video portals and search engines, since they use their own model and dataset to demonstrate on small video sets instead of enabling video retrieval on existing databases. A lot of these systems have been presented in the 90s, like VideoQ [32].

2.4 Video Query Research Prototypes

Video has been studied by different scholar in different linear time and, there are different video retrieval systems developed from earlier to now a day. We discuss different video query research paradigm under this topic. Among different video query prototypes, VideoQ uses visual features for query processing. The query is formulated in terms of element with visual attributes such as shape, color and texture. VideoQ is a web based video search system bases on client server architecture, where queries are detected from animated sketches. An animated sketch is defined as a sketch where the user can assign motion to any part of the scene [32]. Features that are stored in database are generated with automatic analysis of video streams. We elaborate the existing video query languages as follows; content-based video query language [33] and, SQL-based video query systems like BillVideo in [3], which deals with spatio-temporal properties of video data. SQL is extended and used for querying video objects. During query processing each interval video objects are evaluated before query execution. The BilVideo [3] database model uses keywords for storing entities and uses spatio-temporal definition for query processing. Fact-based rules are implemented as Prolog facts in a knowledge-base. When a text query is entered, the query processor divides the query into sub queries, and it fetches the result from knowledge-base facts. Spatial and trajectory queries can be processed using a drawing-sketch tool; user draws the paths or use Minimum Bounding Rectangles (MBRs) to sketch objects. For textual query interface case, BilVideo system uses an extended SQL language. The disadvantage of BilVideo system is user need to learn the details of an artificial language to create rules. Hacid at el. [1] proposed a database approach for modeling and querying video data which works for the basic spatio-temporal queries. The CVQL system [33] supports video object query with spatial and temporal operators using a content-based logic query. Frame sequences which satisfy query

predicate are retrieved with the help of index structure. However, CVQL does not allow topological and trajectory queries.

In most video retrieval system the query techniques determines the query interface type [26], query interfaces generally can be categorized as graphical interfaces and textual interfaces. QBIC-style is a graphical interface system which allows users to query using sketches, layout or structural descriptions. WebSEEk is a graphical interface system which retrieves context-based image/video files from the Web [38]. WebSEEK and SWIM uses sketch query interface which is composed of templates and color histograms. In SWIM user composes units of canvas by specifying visual attributes and, each key-frame template is matched with the frames in the video. Color histogram method works based on modifying the color composition with scenes in WebSEEk. SWIM and WebSEEk uses category or hierarchy terms in video database interface. With the help of sketch pads, user can draw trajectory of the objects in VideoQ [32]. In videoQ the system searches the matching scenes to the trajectory in ranked order. During query by example, users select and ask a video clips that have same motions with input example clips. Text query processing queries uses input questions with qualifiers and listed keywords. Most text query interface uses keywords and those keywords are matched with the indexed video database schema [43]. VideoSTAR is a generic video database especially used for searching TV broadcasting/news and documenting professional archives [46]. In VideoSTAR list of elements and query index terms are choose to formulate query syntax. The most flexible query method is using natural language processing at front user interface in Informedia [45] and VideoQ [32]. In this study works, user does not need to learn any artificial query language. He/she uses his own sentences for query processing. In Informedia the terms are parsed and are tried to be matched with the key frames to video clips.

2.5 Parsing, Event Detection and Refinement

In order to extract information from user query, parsing plays a big role. Parsing is used to process syntactic analyses of a sentence in terms of a given grammar and lexicon. The output of parsing is something logically equivalent to a tree, showing dominance and precedence relation between compositions of a sentence, mainly constituent in the form of attribute-value features [27]. For spatio-temporal query representation, events represent the most role referent elements in addition to time and location. Events are characterized by their participants or arguments

correspond with discourse entities. Since arguments play a key role in describing an event, identifying arguments is useful for finding reference relations among events [22]. Identifying event terms in text query is an essential task for any semantic query system. Events are selected from a list of candidate verbs. Though, action verbs are higher probability of being an event. An event is defined as recursive or complex, if there is an event with other event as an argument. For example, in a sentence "the knife was used for killing the dog" has two events 'used' and 'killing'. The killing event is an argument for used event. Temporal ordering specifies order of occurrence for atomic events in a chain of events [23]. Rowe et al. [31] studied that; video consists of events and instances of activities taking place in video action. For example wedding is an activity, but wedding of Kebede and Aster in a video scene is an event. Events can be thought of as classes, and activity can be thought of as the instances of these classes. The activity verbs can have a number of roles that define events. For example, murder is an activity. In this case murder has two roles defined for murder and victim. The murder of Kebede by Abbebe is an event, where Kebede has the role victim and Abebe has the role murderer. Philips and Riloff [24] proposed exploiting role identifying nouns and expressions for information extraction which exploits the role of nouns with respect to an event. For example, the word kidnapper is defined as the perpetrator of a kidnapping. Similarly, the word victim is defined as the object of a violent event. The proposed approach in [24] identifies noun phrases as a lexically role referent for event description. For event-based information extraction the most reliable pattern usually depends on words that explicitly refer to an event. Generating typed dependency parser from phrase structure [20] presents a dependency relation among phrases in a sentence. Phrase structure represents nesting of multi-word composition and a dependency parser represent dependencies between individual words. A typed dependency parser label dependencies with grammatical relations, such as subject or indirect object. Generating type dependency has two phases' extraction and dependency typing. The dependency extraction phase parses a sentence with a phrase structure grammar in Penn Treebank dataset. But in practice the Stanford parser has high accuracy statistical phrase extraction [27]. Conceptually, the most specific grammatical relation is taken as the type of the dependency.

Carreras and Marquez [2] conduct a research work on semantic role labeling for English corpora. The shared task CoNLL-2004 has more answer on semantic role labeling. To conduct SRL a full syntactic parse is required to define argument boundaries. Recently there are two main English

corpora to train SRL annotation namely PropBank and FrameNet. The CoNLL-2004 shared task address predicates as verbs and it labels core arguments with consecutive numbers from A0 to A5. PropBank annotate the pennTreebank with verb argument structure. The semantic description depends on the verb phrase with a set of tokens. Sutton and McCallum [21] proposed a SRL which provides an opportunity to explore how higher-level semantic information can inform syntactic parsing. The basic SRL system uses argument label for each constituent element in the parse tree generation. SRL requires three stages to preprocess a sentence such as pruning, identification and classification. Pruning refers a deterministic preprocessing procedure introduced by Xue and Palmer (2004) to prune many constituents which have no arguments. In identification phase a binary MaxEnt classifier are used to prune remaining constituents which are predicted to be null. In classification multi classes MaxEnt classifier are used to predict the argument type for the remaining constituents.

Contextual Representation is a major barrier for many systems which accept natural language input. For example, two different senses of an English word form may have different form in annotated database. Therefore, systems for machine translation should be able to determine which sense will go to the dictionary set. WordNet lists alternatives terms which choices must be made based on the target threshold value set. WordNet would be much more useful if it incorporated the means for determining appropriate senses, allowing the program to evaluate the contexts in which words are used [47]. For example, user can ask for a car however, in video database there may not be car entity, but instead Mercedes and Fiat exists as entities. In order not to reply with an empty result set, ontology based query systems are proposed [6, 9].

2.6 Modeling Video Content

Most database systems provide a support for diverse range of applications in multimedia query system. Database management system implements concerning data types including creation, storage, indexing and querying. Having different kinds of data type in video like sound, image, and text we must handle the storage and query issues of video elements. The main characteristics of video elements are temporal and spatial properties. Temporal video data depends on time and spatial data depends on temporal with directional properties. Video is composed of image like structures called scenes [5] with spatial and temporal properties. As we discussed video is rich in information content and this property creates a variety of semantic representation which helps for

better query results [8]. A video data model is a representation of video data based on its characteristics and content, as well as the purpose of applications. Some desired capabilities of a video data model include multi-level video data abstraction, video annotation, spatiao-temporal relation and video data independence. Video data models bases on the idea of video segmentation or video annotation layering [48].

Modeling video data is very difficult due to fuzzy character, because of, inexact boundary of objects and this property creates uncertainties in video data model design. Having these characteristics, a variety of information may be retrieved from the same video data files. Different video data model have been developed for different application requirements. Researchers have done on three basic video modeling methods [6].

Video segmentation model works based on selective features and with their consistence property in space and time. A video stream is divided into video segments using different techniques like histogram matching, algebraic operations etc. Most video segmentation technique works on shot level boundary detection. In video feature space segmentation whenever the histogram changes a new video segment is created. The drawback of this modeling technique is inflexibility [25].

Annotation-based modeling annotates video using keywords or describes attributes with free text. Annotations can be done using keyword-based [5], or using natural language descriptions on video contents [6]. Semantic based content extraction and annotation helps to retrieve textual tags. Semantic extraction is retrieval driven which is generated through user's query. Different people may have various understanding and descriptions on the same video clip and this will affect the retrieval process. Annotating text with objective description with a common content format gives more accurate result. Annotation with multiple tags also avoids possible real-time computation on low-level image features, which fastens the query procedure [30].

Object-based modeling uses object-oriented approaches; this modeling technique bases on semantic representations in video data model design. In this approach video objects are the main interest for modeling. Each video object has a unique identifier, frame number in which the object appears as an attribute set [37]. The proposed [3, 7, 9, 25] approaches helps to index spatial and temporal property of salient video objects. Indexing and retrieving are subject research areas in video data model design. Each video data model uses different querying techniques. Video can be retrieved in two approach ways; the first approach is retrieving video contents on low level features such as color, texture, shape and motion. In addition to low level feature queries, spatial and temporal properties of video objects are further studied in [1]. For

example, “Give me all videos which are similar to the example video with respect to color histogram” this query will be answer based on color histogram matching algorithm. The major drawback of feature-based models is their inability to convey semantic interpretation [5, 18]. The second approach is based on metadata tag descriptions. Low level feature-based modeling tries to model perceptual features that are specific to video content with specific values of color, texture and shape. In general detailed descriptions of image features can be looked to answer user queries. The video content description includes objects, events, activities, and attributes. Spatial property for object is defined with the minimum region which contains the object area. This property gives a chance to find spatial relationships between any two objects. Some of spatial keywords include south, north, west, east, northwest, northeast, southwest, and southeast. Topological relations include equal, inside, contain, cover, and covered by, overlap, touch, and disjoint [7].

Jain and Aygun [17] proposed a video data model that changes high level information metadata to linear string grammar. From the string grammar representation different types of query are generated like event-object location, event-location, object-location, event-object, current and next event, projection and semantic event. The data model identifies components like objects, events, locations and camera views in domain of tennis sport. Object in a video is defined as a region that has semantic meaning and its spatial property changes over time. Objects are main entities that perform the action event. For example, ‘players’ in sport video are objects. Events are main action takes place in video data model. For example, ‘serving’ is an event verb in tennis game. A direction in spatial definition is represented with the space occupied by the object. For example, soccer field is the location for players and ball.

Video is distinguished from other media type due to its temporal and dynamic nature. There are number of research works done so far concerning the description of video object using temporal and spato-temporal data. Such type of queries involves temporal relationships among objects in a video stream. For example, “Give me all videos in which object A appears before object B”. In spatio-temporal query the elements of the data model are objects, activities and events. Spatial property contains location of an object in a video frame. Spatial queries can express simple directional or topological relationships among salient objects. For example, “Give me all videos in which object A appears to the left of object B.” It is more difficult to represent the spatial relationships between two objects in a video than images since video data has time-dependent

properties [6]. Video annotation takes place tagging a free text or attribute/keyword to a video data file. The physical level video segmentation approach does not address semantic concepts such as, objects, events, roles etc. Instead video data is described with a set of metadata set to a specific segment of temporal and spatial properties. Annotating video with free text is easier to retrieve query and describe a video data model clearly [30].

2.7 Semantic Similarity Measures

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 with set of related concept in 'synonym sets' or 'synsets'. The WordNet has an IS-A hierarchy model which have root tree. In WordNet Nouns, verbs, adverbs and adjectives are organized by a variety of semantic relations into synonym sets (synsets). Examples of semantic relations used by WordNet are synonymy, autonomy, hyponymy, member, similar, domain and cause and so on. Some relations are used for word form relation and others for semantic relation. These relations will be associated with words and words to form a hierarchy structure, which makes it a useful tool for computational linguistics and natural language processing [47].

2.7.1 Methods for Semantic Similarity

Determining semantic similarities often comes up in applications of NL processing. The issues in semantic similarity measuring are representation of concept, word sense disambiguation, determining the structure of texts, text summarization and annotation, information extraction and retrieval. The main issue in semantic similarity is getting more accurate results between words for the stored video object and the word used in the query [8]. There have been many methods for evaluating the conceptual similarity, which can be divided into three groups:

Path-based Measures

The main idea of path-based measures is that the similarity between two concepts is a function of the length of the path linking the concepts and the position of the concepts in the taxonomy. The methods in this group depend on counting the edges in a tree or graph based ontology [55]. Finding the shortest path is important, but when the edges are not weighted, like in WordNet, other metrics, such as the density of the graph, link type and the relation among the siblings, should also be considered.

Information Based Similarity

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. The more common information two concepts share, the more similar the concepts are [6 and 54].

Gloss Based Similarity

Gloss is the definition of a concept. Gloss based similarity methods depend on WordNet to find the overlapping definitions of concepts and concepts to which they are related [54]. It has the advantage that similarity between different part of speech concepts can be compared. However, gloss definitions are too short to be compared with other glosses. It is based on the assumption that each concept is described by a set of words indicating its properties or features, such as their definitions or "glosses" in WordNet.

2.8 Natural Language Query Techniques and Video Database

Natural Language Interface has been a very interesting area of research since past times. The aim of natural language interface is to provide an interface where user can interact with database more easily using text to retrieve information from any database systems. There are different works done so far querying video data using NLP techniques, with this NL query we can handle more complex query structures. NLP interfaces in video database have become more sophisticated in order to answer more complex queries with the underlying data model. Syntactic parsers play many roles to convert media descriptions to be stored and build semantic ontology trees. The video data system in SPOT [9] queries moving objects in surveillance videos. It uses NL processing in the form question-answering system; it uses annotation term and works on natural language interface. Phrases extracted in user query are used to describe question types and information segments. In query execution phase user queries are syntactically parsed to match with the knowledge base. When a query match is found between annotated and parsed query phrase, the part of annotated set result is shown as output to the user. Natural language query interface over content-based video data model [6] has the capability of querying spatio-temporal terms, in addition to the basic semantic query. This research work uses natural language interface to answer user query on video database. It uses shallow parser module to parse user

query. Information is extracted to map the parsed queries into the underlying formal query form. Spatio-temporal queries provide the inclusion fuzziness in spatial direction property. The query interface handles ontological based search using domain-independent set. Most natural language query interface systems faced difficulty on correctly matching query with video data descriptions. To solve this problem, main terminologies on entities in video frames/shots are extracted in time bound. When queries are parsed, the first aim is to extract entities that occur in user query and match them with entities in the database but, an exact match cannot be obtained from the query and knowledge base. Informedia uses keyword-based matching system in [19]. These systems have the disadvantage of missing the detailed and more realistic queries since it is keyword based. Most NLIDB systems primarily convert user query language into SQL form [13]. Natural language interface in database is a very convenient and easy method of data access especially for casual users. Natural language use syntactic parser in order to properly relate natural language structure. Syntactic knowledge usually resides in linguistic component. Knowledge about actual database resides to some extent in data model. The success in NLIDB gets more attention due to real world benefits in NLP interface; basically there are four main approaches for natural language interface in database and each approach has its own specific architecture [10].

Pattern matching approach is the first design and development for NLIDBs. It works based on patterns to answer user questions. The main advantage of pattern-matching approach is its simplicity. The approach has no parsing and interpretation modules to answer user query [50], in this system, some patterns are written for certain types of queries and is executed based on the set of rules. For example, if the user asked “What is the capital of India?” using the first pattern rule the system would return “Delhi”. The system would also use the same rule to handle question such as “print the capital of India, “ could you please tell me what is the capital of India?” etc. ELIZA is among the few systems that plays the role in pattern matching approach [26]. However the results of this technique were not satisfactory and new techniques have been developed. The second approach is based on intermediate language, which is used to represent intermediate logical query generated from a natural language query. The intermediate logical query is then transformed into SQL form [26]. The logic is to map a sentence into a logic query followed by the translation of logical query into a general database query, such as SQL. User define the types of domain which refers is-a hierarchy in a built-in domain-editor. Moreover, words expected to appear in queries with logical predicates are declared by the user. Queries are

first transformed into a Prolog-like language LQL, then into SQL. The advantage of this technique is the system generates a logic queries independent from the database schema, and it is very flexible in domain replacements. There are several intermediate representation languages in the process [13]. The third approach is syntax-based architectures, where the natural language question is syntactically analyzed to create parse tree. The resulting parse tree will be used directly to create database queries. Example of such system is LUNAR [19]. In syntax- approach the user's question is parsed and the resulting parse tree is directly mapped to an expression in some database query language. The main advantage of syntax based approach in NLIDB is; it gives detailed information about sentence structure. A parse tree contains a lot of information about the sentence structure like part of speech, how words can be grouped together to form a phrase, how phrases can be grouped together to form more complex phrases until a complete sentence is built. Syntax-based approach uses a language that explains feasible syntactic structures of the user's query [26]. Syntax-based NLIDBs usually interface to application specific database systems. Generally, it is hard to design mapping rules that will map parse tree into some expression directly in a real-life database query language [10]. The fourth approach is based on semantic grammar. This approach involves construction of a parse tree and mapping of the parse tree to SQL. In semantic grammar, the refinement is done by parse the input statement, then maps parse tree into a database query. The disadvantage of semantic approach is it needs a specific knowledge domain, and it is quite difficult to adapt the system to another domain [26].

Table 2.1: NLIDB Approaches

Approach	Approach used	Advantage	Disadvantage
pattern	Pattern based	simple	Used for simple query
Intermediate	Logic based	Easy to translate a sentence into query database	Transformation from logic To query language is not simple
Syntax	Syntax based	provide detailed sentence structure	Hard to design mapping rules
Semantic	Grammar based	simplify parse tree and reduce ambiguity	It needs prior-knowledge/domain

Chapter 3: Related Work

3.1 Introduction

A number of research works have been done in the area of natural language on video retrieval which makes the query processing simple. Natural language interface helps used to query video contents in database interface. Natural language provides a flexible system where user can use his/her own sentences for query request. However, the main issue is translating a given natural language query into semantic representation under the proposed query language. In order to handle these issue different NLP techniques can be employed in order to map queries into database query language. To this end, structural elements of annotated text are obtained by parsing each sentence into smaller token [2].

Video data models proposed in the literature review can be classified in different ways, Due to the complexity of video data. There have been many video data models proposed for video databases in [1,3,7,8]. Some of the existing work use annotation based modeling [30]. Some use physical level video segmentation approach [3,7], and some have developed object modeling approaches which uses objects and events as a basis for modeling semantic information in video clips [25]. The object-oriented approach is more suitable to model the semantic content of videos in a more comprehensive way.

In this chapter, we present the most related works regarding natural language video query processing core points. We also have identified their strengths and weaknesses and, identifying points that make our research different from all those researches will be described.

3.2 Natural Language Query Processing

In this study, natural language is used for querying database interface which provides a flexible system where user can use his/her own questions for querying. When we come to natural language query user does not have to learn any query language, which is a major advantage of NLP. Natural language interfaces provide the most flexible way of expressing queries over complex data models. However, they are limited by the domain and by the capabilities of parsers and the main task is translating user query into desirable high level language form for a better query result.

3.2.1 Natural Language Interfaces over Databases

Natural language play a major role for formal response as its input detected, and takes the parser tools in order to generate adequate natural language response. NL database systems make use of syntactic knowledge and knowledge about the actual database in order to properly relate NL input to the structure and contents of the database. The previous studies of natural language query processing depend on simple pattern-matching techniques. These are simple methods that do not need any parsing algorithm. SAVVY [50] is an example of this approach. In this system, some patterns are written for different types of queries and these patterns are executed after the queries are entered. However, the results of this technique were not satisfactory, because the designed patterns are simple and not flexible to process complex structures.

Another NLP system is LUNAR [51], the proposed approach works based on the parsing algorism and it generates parse tree to formulate syntax rules. The method is especially used in application-specific database systems. The drawback of the syntax based system is the database query language must be provided by the system to enable mapping from parse tree to the database query and, it is difficult to create mapping rules.

The LADDER system [50] uses semantic grammars and the system uses syntactic processing and semantic processing techniques. The disadvantage of this method is that semantic approach needs a specific knowledge domain, and it is quite difficult to adapt the system to another domain.

The proposed natural language query processing system [12] uses natural language interface; and translates user query to a database language query without having knowledge of system language. The spell correction module corrects mistakes made by the user while firing query. The system uses the following procedure to answer user query, first the system accept a string in natural language form and, it will checks the words for misspelled using word pair mining. The system will split each word query into tokens. Finally, the tokens will be transformed in SQL mapping logic. But directly translating a set of tokens into NLQ form will affect the query result.

Kumar and Singh [13] proposed a natural language query interface over video database. The designed architecture combines syntactic and semantic grammar. For example in one of the query “list me all employees”, the syntactic structure identifies terms like, “List me” and it will have verb phrase format and **“**employees” will be noun phrase respectively. The semantic grammar identifies the meaning of its part from sentence structure. The query generator module

maps phrase clauses to structured format. Due to the limitation of directly translating user query to structured format in syntactic model, the intermediate representation model is used. However, the proposed approach lacks to translate user query into SQL form when semantic clauses face in the user query.

3.2.2 Natural Language Techniques over Video Databases

Most natural language querying systems handles more complex query structures. This means that NLP techniques used in video databases should be more sophisticated so that queries can be mapped into the underlying query language. Syntactic parsers can be used to parse the given natural language queries. Query mapping systems used to map queries into their semantic representations, and ontologies can be used to extend the semantic representations of the queries. As a related work, there are NLP techniques in querying video data which convert the media descriptions or annotations and build semantic ontology trees from the parsed query like in Bilvideo and Informedia [3, 45].

The content of video can be queried in several ways some of them are designed for graphical interfaces, form based and textual interfaces. In graphical user interfaces, the user generates queries by selecting proper menu items, sketching graphs and drawing trajectories in VideoQ [15], the drawback of such method is they are not flexible enough to use by user. Bilvideo [3] is a system which provides full support for spatio-temporal queries with any combination of spatial, temporal, object-appearance, external predicate, trajectory-projection and similarity based object trajectory. The proposed systems are built in knowledge base with prolog facts. However the system does not support ontology-based querying and, it is not possible to get close-match results.

The proposed study at [9] queries moving objects in surveillance videos. It uses natural language in the form of question-answering system. The annotations terms are stored in knowledge base. Queries are syntactically parsed to match with annotation terms. When a match is found between annotations and parsed query phrases, the result is displayed to user as a segment query. The representations are in sense of matching the annotations in knowledge base. But it works on simple query type.

Natural language based query using video database [4] proposed that the part of speech tagger is applied for extraction of clauses from user query. The approach uses frame based and can query object, spatial and similarity based object trajectory queries, like above, right or left. The

proposed approach specify user queries as object-appearance, spatial, and similarity-based object trajectory queries based on ordering of POS tag information. Nevertheless, they don't work on temporal and predicate based queries. The extracted clauses from the user query are compared with stored tagged pattern and, execution starts when a match exists. From this research work we can query simple and complex query like "The bird is inside the cage and the house" and, the proposed approach answer user query based on tag order which directly used in query execution process. But in our case we will refine verb and nouns to get desired query result. In additions the proposed system works only on known spatio-temporal operators and, comparing our work, we add different NL operators beyond the listed operators in these works.

Conversation based natural language interface using relational databases [10] aims to use pattern match by integrating goal oriented conversational agents and knowledge tress. Knowledge trees are used to structure domain queries. The conversational dialogue system interacts with user turn by turn in text form. The proposed architecture has four components such as conversational agent, knowledge tree, conversation manager, and relational database. The conversation manager handles communication among all components. Creating SQL query form is created by combining knowledge tree and conversational agents. For example for the query "I want sales for all products in the current month", first it extracts clauses, maps user query to structured language form and finally display the result. But the proposed approach use simple English queries for searching results.

Erozel et al. [6] proposed a video retrieval system using NL query. The basic query supported in this work bases on occurrence queries with a combination of objects, activities and events. The WordNet database holds semantic relations between terms. The query processing maps NL queries into their semantic representations using link parser and an information extraction module. The link parser extract information from the input query and, the information extraction module create a semantic representation. Semantic representations typically has two phases, first it checks what types of query asked and, second which parts of the query elements correspond to the parameters in the data model. Similarity measure between query objects and objects in video database are measured based on a distance metrics. The proposed approach is restricted to have one or two parameter respectively for query representations. From this we can infer that the approach can't answer more than two parameter values. The retrieval works on key frame level with no consideration of shot dependencies and scene concept.

Liu and Li [30] proposed video segmentation process which involves boundary detection between uninterrupted shots and scenes. Important features from key-frames are extracted and denote them as indexes; low level features are motion, color, shape, texture and color histogram, etc. Textual tags are added automatically to low level features in relative of temporal position. The proposed system answer queries if a match is get from the stored textual tags. For example for query "A car is running by a tree", the system checks the existing tags for the query and if they have the same semantics interpretation the query result will be post. But the proposed system only answer simple queries based on occurrence of object tags.

Lee et al. [22] proposed the presence of textual captions and audio in video frames enables building automated video retrieval systems. Accessing content-based video retrieval requires structuring the content and require good query interface. Text detection in video involves detection, localization, enhancement and recognition, to do this region of spatial domain is used. In the proposed approach extracted entity features are stored in video database. The entire video is processed frame by frame for locating textual blocks. Video segmentation is carried out when changes occur in color/motion statistics corresponding to the text regions within time. User can search related videos when a match is present in video database. However, it works on a single visual attribute features (keyword based query).

3.3 Video Data Models

Video data has a rich information source with different data types. The basic elements of the data model are objects, activities and events. Due to this, a variety of information may be retrieved from the same video data. Different modeling techniques on video data have been developed for different application requirements.

The proposed video data model in [5] describes video schemes in three ways: frame-based scheme, the object-based scheme and the external description block. The study work focus on frame-based level scheme and with external description block. A key frame in scene-shot-key representation uses image models under OR-DBMSs environment. The proposed external blocks and objects are annotated with keywords and attributes with free text. Comparing our work to this research, we use NL and we design our query representation model to retrieve a more complex query result [3]. Since this system does not support ontology-based querying, it is not possible to get close-match results.

There are lots of research works on modeling spatial and temporal contents of video. Jain and Aygün [17] developed a spatio-temporal querying of video content through SQL using video database. The data model design uses event and object terms for spatio-temporal definitions. The contents are represented in a linear string with the help of grammar in G-SMART tool. The linear string representation grammar defines rules for representing video contents. Spatiotemporal information are extracted and parsed with the help of grammar rule. The grammar describes and represents various spatiotemporal rules. However, compared to our work the representation of data elements and grammar rules are not flexible; the approach is domain-dependent and relies on a set of defined template representations.

In video data model design, Koprulu et al. [7] proposed a spatio-temporal querying in video database that uses video data entities such as activities, events and video objects. The video data is divided into fixed temporal time sequences. The proposed model supports querying spatial properties of objects and spatio-temporal relations between objects in video frames. They use AVIS data model as a base study, since AVIS models objects which have no attributes other than role in the event. But the proposed approach works on simple query and works on video frame level.

Content based video retrieval [14] proposed a segment of video frame from the extracted object and entities. Extracted video frames are classified as low-level and higher-level features. Low level features such as color, texture, shape, motion, object, face, audio, and genre are used for indexing and retrieval purpose. High-level semantic feature extraction has been surveyed to cope up the semantic gap among content based features; object ontology's are proposed to generate meta- data to define high-level concepts. Machine learning tools extracts low-level features and generates high-level concepts based on color histogram and texture features. The semantic metrics is done between the stored feature in database and user query. The smaller the distance the more similar frames are retrieved. Compared to our work the proposed approach works on low level visual features and, most of user query would prefer to use text instead of visual attributes.

Video scene retrieval through online video annotation [16] addresses how video scenes are retrieved from the associated scene tag keywords. The created tags are classified into nouns, verbs, and adjectives. To get high quality tags they use screen tag selection using automatic and manual process and this helps to remove tags which have no association to the video content.

During the process of scene tag extraction users select appropriate scene when tags appear in time code frame. Retrieving video scenes in this approach will requires submitting an arbitrary number of tags as a query. The approach uses keyword based system.

3.4 Summary

Query formulations give a standard query format to query the video contents with a set of objects and events by defining spatio-temporal operators between objects. Video is hierarchically divided into scenes, shot and frame level and most queries rely on key frame level metadata. Most existing works on NLIDB [6, 10, and 12] uses frame based video retrieval.

All the works in video retrieval uses object, activities and event data model for query processing. Most of the query processing uses annotation text or content based format. Video annotation helps for creating a formal statement which is built to describe video scenes. This formal statement can easily specify what is happening with video and, it can state cause and effect relationship between objects and events. This helps to clearly understood by any viewers of the video retrieval users.

In this chapter, we have reviewed different works done so far using natural language interface over video database. The related works are classified as content based video retrieval, natural language interface for video database and annotation based video retrievals. Content based video retrieval uses low level visual features like color, texture, shape and other features from the extracted image. In content based video retrieval approach the search analyzes the content of the image rather than the metadata associated with the image whereas, NL interface works on associated metadata over the image. Most video database system stores video information in a content-based or annotation based data models. Natural language based video retrieval approaches proposed in the literature [6,10,11,12] works on simple query and the approaches can't process and detect complex user queries. In natural language query, similarity dependency and relatedness has to be done among objects found in user query and objects found in the database. Hence the metrics gives video scenes which are associated to the video contents. In any videos, shots in a scene are dependent on one another concepts and highly define what happening in the current shot as well as objects is found in the representative key frames. As a result, video information at frame level can't give complete information for the entire videos. NLIDB system process and extract objects and events from the user query, but the extracted terms are directly translated to a SQL query form without refinement has taken place.

Chapter 4: Natural Language Based Video Query

4.1 Overview

In this chapter, we present the general architectural design of the proposed natural language based video query. We discuss the detail description of the proposed general architecture of a natural language query from the annotated video database perspective; which works based on the prepared annotated video scene data set. We also present components of the architecture with their detailed functionality and the algorithms developed to retrieve a complete and satisfactory query result in organized format.

The proposed NLP based video query system formulates standard query format and it gives a video result at scene level. Converting user query to equivalent query form needs exact operator definition and set of logical operators and, this helps to get precise answers. Apart from getting exact match we also add similarity metrics not to retrieve null result. In this work, the query process identifies different spatial and temporal operators for query generation and it sends video scene as final result.

4.2 Modeling Annotated Video and Representing Natural Language

Annotating video with text gives a higher description than low level feature descriptions since the semantic descriptions at higher level fills the gaps in video data model. In our scenario case, having high accuracy annotated video data set helps in improving the performance of video query retrieval. To demonstrate the detailed description we have chosen soccer video annotated dataset. The choice is made based on factors such as availability of sufficient resources on the web and it has high semantic information practice to show. Representing soccer video in text form enables to answer different types of user queries. As we know most non-technical users will not be able to access the database unless he/she knows the syntax and semantics of firing a query to the database system. But using NLP, this task of accessing the database will be much simpler. User query will be processed using different NLP tools such as Stanford parser, openNLP etc. Most of user queries can be expressed using SQL and few more complex operators. Finally, our proposed video query gives a video scene which is meaningful and relevant to user query. We explain each of these with the example on soccer video in the following subsections. Figure 4.1

represents the main components of the system together with their subcomponents or modules, and the interaction between each of the main components and their subsystems.

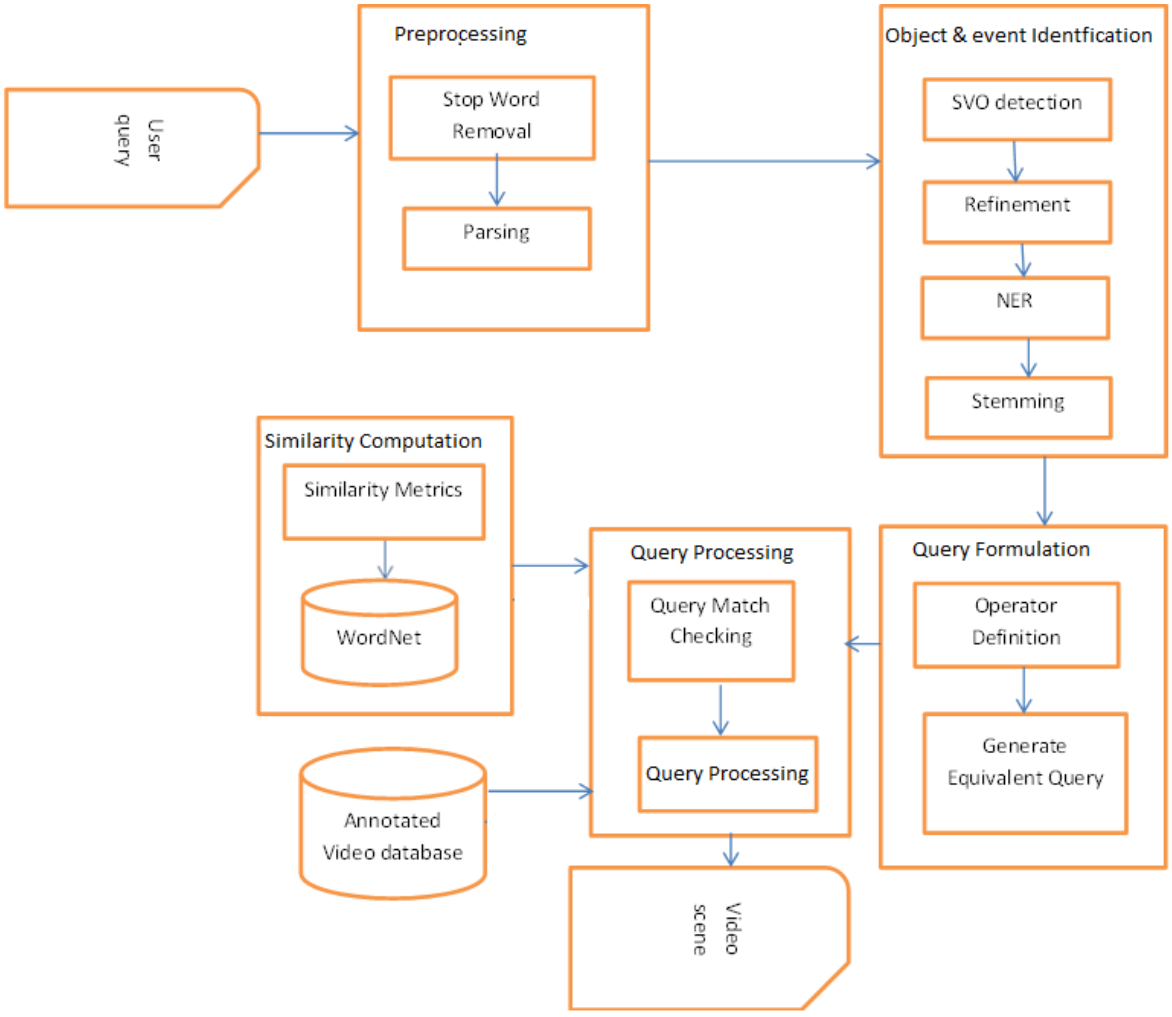


Figure 4.1: Proposed Architecture for NLP based video query

4.3 Components of the System

As discussed in Chapter 2, query in natural language requires syntactic structure and operator refinement as a primary task. This helps to get a better candidate verb and noun result. Query formulation rules and syntax generation have higher importance of quality query result. Thus, our proposed video query can return relevant answers for spatio-temporal and predicate based questions in terms of video scene level. Moreover, our object detection method uses directional relations which are defined using Allen's temporal interval algebra. The coordinate point of MBRs is the relative positioning of its objects with respect to each other. Each components of

natural language query processing which shown in Figure 4.1 are clearly described in the below section.

User query are input in English language, without any special syntax or form. During writing a query user has to include at least an object or an activity terms which helps for better query process to get relevant output. If the basic terms are found initially, then the parameters that include one or more basic words can be expanded. If the user query does not include terms containing either of them we suggest appropriate terms from the dictionary which are similar to user query interest. Before we describe the detailed description of query representation, we will elaborate the query preprocessing process.

Preprocessing is done before feeding all query statements to query engine. For example, the English query phrase may have irrelevant terms which have no impact on query generation like stop words, punctuation etc. The query processing phase does not need all of these data and it should be preprocessed before feed to the query execution. Only the relevant entities are extracted and fed into the query engine. The query preprocessing module is responsible for tokenizing the English queries into words, eliminating English stop-words (less informative words), and stemming inflectional and some derivational English morphemes. The preprocessing stage can investigate different sub components like tokenization, stop word removal and stemming. The detailed description of this component is presented below.

Tokenization: this component splits the English query phrase into English words using white spaces and some English punctuation marks as word delimiters.

Stop words removing: during user query we may face terms which are not relevant for the retrieval task. Those terms include articles, conjunctions, prepositions, etc. For example, words “a”, “an”, “are”, “be”, “for” are referred as stop words. The techniques used to remove these terms differ from system to system and depends on the goal of the query system. Thus, this subcomponent is responsible for removing stop words for further processing.

Stemming sub component process terms which cannot stand alone like suffixes and prefixes which collectively known as infixes. The suffixes and prefixes form inflectional and derivational morphology of the language. Past tenses of regular verbs and plural forms of nouns are examples of inflectional morphology which inflects or alters the word by adding the suffixes.

The stemmer component helps to generate the root words for each identified tokens since, we use a database to store objects and the stemmer helps to better model user queries from the stored objects.

The parsing component analyzes user query by taking each word and determining its structure from its constituent parts. From the parsing component we get subject, verb and object terms. The parts-of-speech tagger assign word label to its linguistic parts. Parts-of-speech include nouns, verbs, adverbs, adjectives, pronouns, conjunction and their sub-categories. For our thesis work we extract word groups like objects, activities, spatio-temporal and start of the interval to obtain desired syntax generation. For this implementation, we use Stanford parser tool which is a more robust and gives high accuracy parsing result.

Event object detection extracts event and object. Subject, verb and object detection in linguistic preprocessing indirectly helps us to extract event and object terms, since subject and object are indicative of object and event respectively. Events can be expressed by verbal predicates and with their arguments. List of verbal phrases which are extracted from the part of-speech tagger are [VB], [VBP], [VBG], [VBD] and [VBN].

Nouns are expressed by noun phrase and for extracting nouns we use named entity recognizer. List of nouns which are tagged in query are [NN], [NNS], [NNP] and [NNPS]. Depending on the extracted SVO, the refinement component will show weight order for term candidates based on in Algorithm 4.1. Algorithm 4.1: An Algorithm for Object and Event detection

```
Input: POS tag words
Output: object and event terms
Begin
For each POS terms
  If POS contains noun phrases
     objectSet.add(nouns(N))
  Else If  POS contains verb phrases
      eventSet.add(verbs(V))
    Else
    Try for remaining words in the query
End for
```

Event object refinement extracts main entities from the extracted candidate terms. The basic approach involves finding event-indicative seed words. The parser module tags the verbs of the sentence with a 'v' tag, but all of them are not main verbs and all of them do not require subjects. Here, a main verb is considered to be the word in the verb phrase which actually represents the action role. In order to identify the main verbs, all the tokens from the parsing result which are tagged with 'v' are considered first. To select main verbs, the Stanford dependency parse tree helps to create the dependency relation among the words in the sentence. The Stanford dependency parse tree usually starts with the main verb at the root of the tree to determine whether a base form verb should be tagged as VB, VBD or VBP. All the other words are then fallen from it in terms of their relation to the main verb to create a parse of the sentence. So, if we parse a sentence using a dependency parser, we should then be able to examine the root of the parse structure returned to see if it is the main verb. In additions, verbs in the infinitive mood are used as parts of speech more than verbs. It expresses being or action. For example, 'Messi kick the ball to the left of Fabregas'. If none of the conditions is met from the refinement module in the above scenario we try for the remaining words in the query.

The refinement module extracts different type of operators in relation to the spatial, temporal and natural language terms in the user query as described in Algorithm 4.2. Listed operators are attached in Appendix B.

The relationships among extracted object terms in user query and the operators help to fill the gap during query process. Having these operators' we also keep the precedence of the operators in accordance of the argument they have. The extracted operators help to identify a valid argument relationship among the video objects. For example 'find videos Messi touches ball' from this query, the term 'touch' is a spatial operator which describes two object regions are contact each other. The results of the refinement operators are further enhanced in the stored procedure component to define the behavior of the operator in a clear way. From those extracted operators we can able to formulate a sound syntax query formation, which retrieve a more complex user questions. Because, the query holds information relation to the operator's scope, number of argument it accept and the return value which are defined in the procedure syntax. In general, answers to compound queries are found by directing the output of one function as input to another function. Let us suppose that a user query wants to find the parts of a video clip

satisfying the following conditions: Query: “Abebe kicks a ball before kebede”, where Abebe, kebede and ball are objects and kicks is a candidate verb. The temporal operator “before” returns a ball kick before Kebede, the result comes true if a valid relation is set from the stored procedure rules. Spatial operators are defined and become the parameters for the listed objects and extracted NER during in query representations. Some of the supported spatial relations are above, right, below, north etc. If a user query contains terms containing a spatio-temporal operators the system will check in the database in stored procedure module, if it is true then, it will be added to the query processing stage. If not, it will back again and do the previous refinement and, it will combine to the basic logical operators.

Algorithm 4.2: Operator refinement

```
Input: SVO

Output: spatial and temporal operator

Begin

For each SVO

   If SVO contains spatial term

      SpatialOp.add(term)

   Else If SVO contains temporal term

         temporalOp.add(term)

       Else Try the refinement again

   Else If SVO contains logical operator term

         logicalOp.add(term)

       Else Try the refinement again

End for
```

4.4 Data Model

For data model design we use objects, events, and activities which, are extracted from the user query. The video objects may be a named entity collection in a movie such as, James, John, etc.

or they may be objects such as a table, ball, etc. An activity term is accepted as the substance of a given frame sequence. Since a video frame gives a complete description of objects which are detected it the key frames. Activities are the type of the action performed in a video (for example walking, throwing, kicking ball). Multiple activity types may occur simultaneously in a frame sequence. An event is an instance of an activity type. It consists of a unique activity type, a set of roles in the activity, and objects as the actors of roles in the activity. For example, in the event query ‘‘john is kicking a ball and peter appear to the left of ball”, the activity type is kicking and left, the role is kick and left, and the actor is john and peter, and the object is ball.

As we described in Chapter Two, many video applications uses temporal, spatial or a combination of this to answer user queries from the video repositories. Some of them use annotation to overly the content information on top of the video streams. Each annotation is associated with a logical video segment, which is, in general, a subset of a video stream and is defined by the starting and ending frame numbers. Our video data model supports different queries based on the number of arguments and the availability of operators. Our query language uses logical operators, spatial, temporal, predicates and natural language term operators to generate standard query format. To model time we use a point-based where the object annotation lays in video frames in scalar time representation. In our data model for spatial operator definitions we use rectangle coordinate points and, for temporal definition we use Allen's interval operators such as overlap, precede, contain, equal, meet, and intersect which are the basic temporal operator. The commonly used spatial predicates are overlap, meet, contain adjacent and common border, etc. Spatial operations include intersect, area, distance. For instance, to retrieve a video scene where annotations are set for a given objects, the query processor has to check that there is a temporal or a spatial relations between the individual object representation.

4.5 Query Representation

Since we represent the video content as a spatio-temporal term, due to this representing the video content as a text helps user to describe many spatio-temporal queries. For object annotation in video we use the two dimensional coordinate system, in which the spatial property of the object is defined as object’s regions appearing on the screen. We have used the imaginary rectangle points in approximating the regions of object which covers all parts of the objects. In our model, the specified rectangular areas on the screen represent the locations of objects. When we come to

the user query we may have atomic structures such as one event or object attributes, and some of them may have a relation between any two or more object and event entities, in this case we identify the operator which creates a relation among the entities and what dependency is in the query sentence. From the metadata description we extract information like Hierarchical number, entity name, detailed description of the entity, the number of operators to which object that are allowed. Datatypes can also be extracted whether the element's value is textual, numerical or a Boolean, and any constraints on its size and format. In our query representations user queries are expressed using SQL and few complex queries; this can be written by extending SQL through more spatial and temporal operators. AND, OR and NOT are logical operators we usually use them to answer question which have more alternative entities in query processing. AND operator represents the combination of a set of sub-entities to form composite entity, this operator gives a valid result only when all the condition set are true. OR operator represents the set of alternative compositional entity set.

In our case, we use all logical operator to formulate sound query structure because, in our query representation we have a set of different query representation with set of sub queries so that logical operator helps to connect segment queries for query formation process. We will describe the query representation scenario for extracted object, event and spatiotemporal data element in in Algorithm 4.3.

For example in soccer game objects are ball, players etc. are represented as <obj > = <player>. Events are identified as candidate verbs and the main events are represented as E=Shot, kick, dribble and so on. Spatio-temporal is represented in accordance of directional, topological relationship like, above, below, meet etc.

For the above description, we use SQL schema definition in order to evaluate query process.

```
SELECT <items>

FROM <tables>

[WHERE <search-condition>]

ORDER BY <time period>
```

Algorithm 4.3: Query representation

```
For each natural language query posed
  Check for question particles right,left,below,equall…
  If question contains one of these question particles
     Classify question as spatial type
  Else If question contains before, after, between...
       Classify question as temporal
  Else If question contains terms pass, throw, kick...
          Classify question as predicate operator
  Else If question contains of, score, against...
          Classify question as NL operator
  Else Try query formulation for the remaining terms in queries
End for
```

4.6 Spatio-Temporal Definition

In a given video scene, the spatial relationship between two objects can be defined using the spatial relationship between the minimum bounding rectangles of each object. This property gives us the ability to find the spatio-temporal relationships between any two objects in a frame sequence. We use the spatial properties to objects to define the spatio-temporal relationship in a video stream. The spatial predicate relations can be directional and topological. Directional relations include south, north, west, east, northwest, northeast, southwest, and southeast. Topological relations include equal, inside, contain, cover, covered by, overlap, touch, and disjoint.

According to our definition, interacting objects can have directional relations associated with them as to the case where Allen's temporal interval algebra is used to define the directional relations. In order to determine which directional relation holds between two objects, metrics of coordinate points in objects are used. However, if the coordinate points of the objects' are the

same, then there is no directional relation between the two objects so we interpret as the two objects are equal. Otherwise, the most intuitive directional relation is chosen with respect to the min and max values of x and y. In the example given in Figure 4.4, object 1 is to the *left* of object 2 because the coordinate points of object 2 in MBR are greater than the directional line to the object1. The MBR enclose the object region to find the object location in video frames. The computer vision tools like opencv provides a method of filtering coordinate points. In our case we use MBR to take the pixel coordinate values as shown in Figure 4.2.

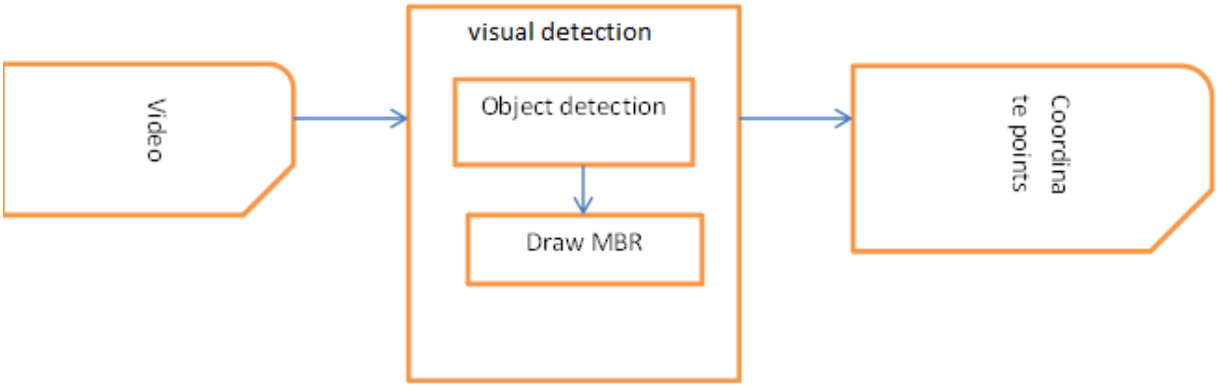


Figure 4.2: Taking object points

Definition: the directional coordinate points of object1 are defined the inclosing coordinate points as the max and min pixel values. Object is defined as a set of points, object= {minx,miny,maxx,maxy}

When the directional coordinate point of MBR are identified, then the directional relation between objects 1 and 2 is easily defined with mathematical operation where object 1 is the one for which the relation is defined. The rest of the directional relations can be determined in the same way as shown in the Figure 4.3.

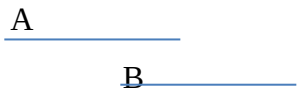
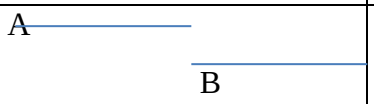
	A *OVERLAPS* B
	A *CONTAINS* B
	A *EQUALS* B
	A *MEETS* B

Figure 4.3: Example of Spatio-temporal definition

We define the following basic spatial predicates with their linear arithmetic constraints. Based on the Figure 4.4 definition;

Example, a directional relation for two objects is the location of an object1 is defined by the rectangular area (a region) where object1= $(X_1, Y_1), (X_2, Y_2)$ where $X_1 < X_2$, $Y_1 < Y_2$

Figure 4.4: Minimum Bounding Rectangle - MBR

Left (point, object): a point $(x_o; y_o)$ is on the left side of a region in rectangular area $((x_1; y_1); (x_2; y_2); ((x_3; y_3); ((x_4; y_4))$, is an edge of a rectangle for two objects.

left(): $_{x1}, x_2 > x_o, y_o$ and $x_3, x_4 > x_o, y_o$

Right (point, object): a point $(x_o; y_o)$ is on the right side of a region in rectangular area $((x1; y_1); (x_2; y_2); ((x_3; y_3); ((x_4; y_4))$, which is an edge of a directed rectangle, is

right(): $x_1, x_2 < x_o, y_o$ and $x_3, x_4 < x_o, y_o$

Before (object1, object2): an object1 is before object2 in region in rectangular area (region), if the below condition met;

before(): $(x_2,x_4)<(b_1,b_2)$

After (object1, object2): an object1 is after object2 in region in rectangular area (region), if the below condition met;

after(): $(x_2,x_4)>(b_1,b_2)$

Inside (point, object): a point is inside a region if the point is on side of all four edges of the MBR. From the Figure 4.4 we define the relation set as follows.

Inside(): x_1,y_1 contains x_o,y_o and x_2,y_2 contains x_o,y_o

Inside(object1,object2): an object1 is inside a region in object2 if the object1 region is on side of object2.

Inside(): $(b_2-b_1)/(a_2-a_1)=(y_2-y_1)/(x_2-x_1)$

Equal (point1, point2): a point1 is equal to point2 if the set of the end points of point1 is equal to the set of the end points of point2:

equal(): point1=x_0,y_0 and point2 x_{01},x_{02} are equal iff $x_0,y_0= x_{01},x_{02}$.

During (object1, object2): an object1 occur during in object2 if each regions coordinate points meet the below metrics.

during():$(x_1,x_2)<(a_2,b_2)$ and($y_1,y_2)>(a_1,b_1)$

Overlaps (object1, object2): for two regions, object1 $(x_1,y_1),(x_2,y_2)$ and object2 $(a_1,b_1),(a_2,b_2)$ is:

overlaps():$(y_1<b_1),(x_1<a_1)$ and $(x_2>a_1),(y_2>b_1)$

Meet (object1, object2): if one edge of object1 is on an edge of object2 and the rest edges of object1 are not inside object2 is:

meet():$(x_1,y_1),(x_2,y_2)=(a_1,b_1),(a_2,b_2)$ and (x_2,y_2) or (x_1,y_1) # (a_2,b_2) or (a_1,b_1)

4.7 Query Formulation

The proposed query formulations handle spatio-temporal definitions in specified query forms. For more complex query types we segment user query into sub query and finally we combine the sub query result as a final query formulation form. The completed query formulation syntax contains all the specified sub query properties as each object appears in each segment queries. The query formulations rule creates sound syntax by combining sub queries with logical operator and spatio-temporal operators. All temporal and logical operators' definitions are binary and, if

more than two sub queries are given as arguments to binary operators, the first two are processed first and the output is pipelined back to the operator to be processed with the next argument. We define operators to each sub queries and, new query is augmented to the list in hierarchical form. The formulated queries are loads to the query engine to get final video results. Furthermore, any sub query may also be sent to the query server at any time to obtain partial result if requested. Let us suppose that *a* and *b* are two salient objects and that their spatial relations are denoted by *inside(a* and *b)* respectively and, let us also assume that *B*1 denotes a temporal sub query on objects before *b*, the query formulation module will give as inside(*a*,*b*) AND before(*B1*,*b*) forms. The individual operations performed are appear(*a*) AND appear(*b*) AND inside(a,b) AND (before(B1,*b*)). After the sub queries are formed, the final query can be substituted in the below query template form. The query template form is Query=[spatialOP^{+} NE [and NE[obj]]* and temporalOP NE[obj] [and NE[obj]]*].

We now apply the above pattern representation for the following scenario query cases. In our case a complex query is a parameter query which searches using more than one entity description and the entities may have multiple interconnected values. Our NLQVR system supports an efficient and effective way of formulating user query using SQL to retrieve a video scene from a large collection of video as well as similar videos in database.

Let us consider a query to retrieve a video scene where Messi appear to the left of ball, we represent a query sequence as follows. Result output from refinement module is: obj=Messi and ball and operator=left. In the above query template form we generate Query=(left(Messi,ball)).

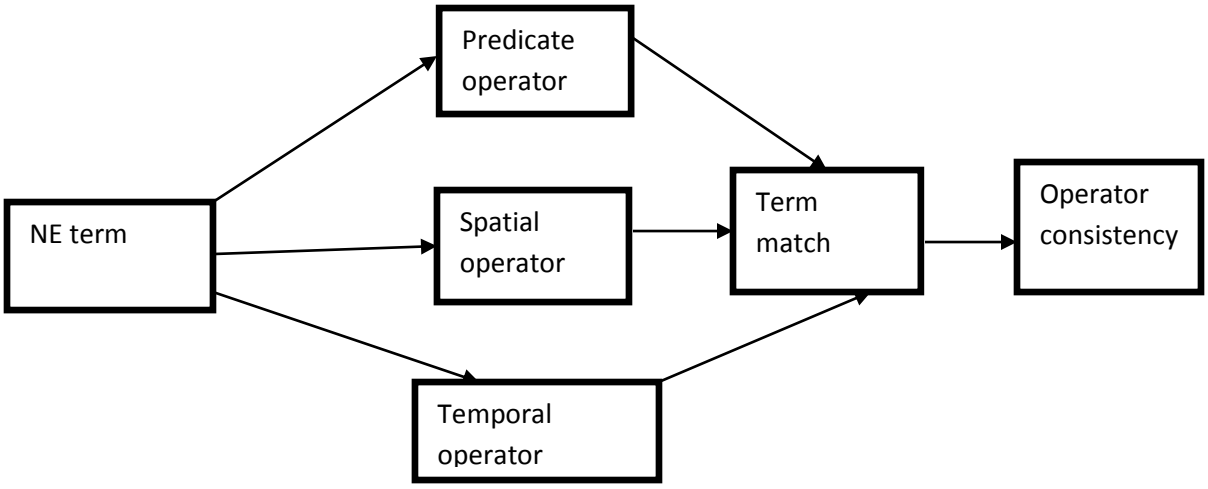


Figure 4.5: Query representation

Based on the query representation in Figure 4.5 we can generate a formal query template form which helps to create sound query syntax for user queries.

Query=[spatialOP NE [and NE[obj]]* AND|OR|NOT TemporalOP NE[obj] [and NE[obj]]*].

Query=predicateOP(NE,Obj)*. User stated object terms are considered as arguments after each operator are detected. The consistency pattern helps to initialize the available operators if the first operator's definition comes true. Let us check the above query pattern for the following scenario cases For example, "Messi appears before Ronaldo and Ronaldo to the left of ball" in this query before is a temporal operator which takes the two arguments NE Messi and Ronaldo and left is spatial operator which shows direction comes after the object ball. Our query form represents like, Query=[before Messi [and Ronaldo] and left Ronaldo[and ball]]. Based on the above query form we define different query representation scenario cases as follows.

Spatio-temporal-Object: in this pattern representation the object is found in spatial and temporal operators. For example, in a soccer game 'the goal keeper to the left of Messi and before ball' here, the 'goalkeeper' is the object and 'left' is the spatial term and before is temporal operator. Since for complete query representation we have <Q> : <spatio-temporal> <obj>. The pattern to be searched is a sequence of object and spatial. This template structure can be written in the following query form: Query = [spatialOP NE [and NE[obj]]* AND|OR|NOT TemporalOP NE[obj] [and NE[obj]]*].

Spatial-Object: spatial-object queries retrieve a video scene containing spatial and object occurrence in video database. For example, in soccer game 'ball pass over the bar in north direction', 'ball' is the object and 'north is the spatial. Retrieval of such video sequence is called spatial-object query. From the full query representation we can ignore temporal instance and the pattern to be searched will have a sequence of spatial and object. These queries can be written in the following way: Query=[spatialOP NE [and NE[obj]]*.

Temporal-object: temporal-object queries retrieve a video scene containing temporal and object occurrence in video database. For example, in soccer game 'the ball appears during in 5 second', the object is ball and 'during is the temporal operator. Retrieval of such video sequence is called temporal-object query. From the full query representation we can ignore spatial instance and the pattern to be searched will have a sequence of temporal and object. These queries can be written in the following way: Query =TemporalOP NE[obj] [and NE[obj]]*].

Query match checker: queries can become quite complex especially when combined with different sub queries. This becomes a bit difficult to check whether a query is valid or not. Having this the query engine directly accept the formulated query to the query match checker component, the query match checker loads the formulated query and directly maps to the underlying annotated video database, if a match gets from the database, the result set will be sent as a final output and it will be displayed to the user. If a match is not get from the database, each entity set will be measured in similarity analysis component along with the WordNet dictionary. In the design of video data model, each type of spatial and temporal operator have their own function definitions and, these functions definitions accept the parameters as the number of arguments is fulfilled. The created query representations can be mapped to each function as the condition is satisfied. Finally the query matcher returns the query results. If the parameters of the functions definitions and the query representations are the same, the appropriate function is called depending on different case operator definition. Finally, for each entity occurs in the query representation, the function query is mapped and the query is executed.

4.8 Semantic Query

The similarity metrics computes the similarity between terms found in user query and the stored database objects. Queries written by user may have different names but could be semantically similar; this could be caused due to differences in naming of an object. The similarity measure handles such difference using WordNet database dictionary. For example, user query may contain term “ball” but this term may have been described in database in different term having similar concepts. The query representation creates an equivalent query which contains object terms during user query, then this objects are mapped to the database for exact matches. The semantic query which holds objects are taken as an argument for operator functions, if exact match is not found the similarity metrics is expanded with new words that are semantically similar to the words appearing in queries. The similarity algorithm finds the words that are semantically similar to a query word using the WordNet dictionary. Whenever the query includes a word representing an object, or an activity, the similarity algorithm is invoked for that word, in order to get similar terms representing objects or activities in the video database. Finally, the query engine retrieve similar video scenes if the object property meets the requirement based on the operator definitions.

4.8.1 WordNet Ontology and Similarity Measure

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. Ontology consists of concepts, attributes, and properties representing relationships between concepts. Ontology properties represent user-defined relationships in hierarchy level form. For instance, the concept 'person' has a synset of {person, individual, someone, somebody, mortal, human, soul}. All these words can represent the concept 'person'.

A distance based similarity method is used to find the shortest path in WordNet which has many inheritances between the synsets due to multiple hierarches. Wu and Palmer's conceptual similarity works based on the factor of the path lengths by using IS-A hierarchy. It assumes that the similarity between two concepts is the function of path length and depth in path-based measures.

The similarity metrics as shown in Algorithm 4.4 works based on finding the query word and an object in video database. The WordNet evaluates the semantic similarity degree between the query word and the object for all sense pairs. The highest sense similarity value is taken as the video object for the query word. This operation is done for every video object which is found in the database. The similarity values are sorted in descending order and, the corresponding video scenes are returned in accordance of the time order.

Algorithm 4.4: similarity measure

```
Input: object term

Begin:

For every  object

Initialize WordNet Database

Get the synsets for objects.

  for each synsets

    calculate relatedness score of synsets.

       If(score of synsets > threshold value)

          Add synset to the query process
```

```
       Else

          Try for the remaining synsets

Output: max score for synsets.
```

4.9 Query Engine

The query engines load the formulated query and execute on annotated video database. The ultimate goal of query engine is to display a list of possible answer without missing the video scenes. The formulated queries are transformed into SQL query engine before being sent to the video database. Each operator definitions call the stored procedure and waits for query answer. The query engine returns a set of video scenes if there is a match. If not, the similarity metrics will take over the actions. The query engine groups each operator into sub queries and, each sub queries are processed separately. Finally, each result is combined and we apply logical operator to get final video scene**.**

4.10 Supported Query Types in the System

Object Queries

Object queries used to retrieve salient objects in user questions. Each video scene is retrieved which satisfies the conditions along with this criteria. Some example queries of this type are given below:

Query: "Find all video scenes from the database in which object o1 appears."

```
Select videoscene
From all
Where appear (o1)
Order by time;
```

For example, find video scene where ball appears in the court.

Spatial Queries

Spatial queries used to query videos by spatial points of objects defined with respect to each other. Supported spatial queries are like directional relations that describe order in 2D space, topological relations that describe neighborhood and incidence in 2D space. There are eight

distinct topological relations: disjoint, touch, inside, contains, overlap, covers, covered-by and equal.

Query: “Find all video scenes from the database in which object o1 touch o2.”

```
Select videoscene
From all
Where touch (o1,o2)
Order by time;
```

For example, find video scene where Fabregas touches ball from a collection of videos.

Temporal Queries

Temporal queries are used to specify the order of occurrence conditions in time. Temporal operators process their arguments only if they contain intervals. Most video query language supports all temporal relations as temporal operators are defined by Allen’s temporal interval algebra. Some of the supported temporal operators are before, meets, overlaps, starts, finishes and their inverse operators.

Query: “Find all video scenes from the database in which object o1 appear before object o2.”

```
Select videoscene
From all
Where before (o1,o2)
Orderby time;
```

For example, find video scene where Fabregas appear before Messi from a collection of videos.

Spatio-Temporal Queries

Spatio-temporal queries contain combinations of temporal and spatial property for each set of objects. Spatio-temporal embodies spatial and temporal entities, and captures a spatial and temporal behavior which deals with geometry changing over time.

Query: “Find all video scenes from the database in which object o1 appear before object o2 and object1 meets object3.”

```
Select videoscene
From all
Where meet (o1,3) and before(o1,o2)
Order by time;
```

For example, find video scene where Fabregas touches ball and Messi appear before Fabregass.

Semantic Queries

Semantic queries used to find video scenes with semantic features in the video repository. The similarity algorithm finds the semantic distance between a query word and a stored video object using WordNet. Since the query word and the video object word can have many word senses, we find the similarity values for all sense pairs. The sense pair with the highest similarity value is taken as result video objects. This operation is done between the user query and object names found in the video database. It is more discussed in Chapter Five.

Query: "Find all video scenes from the database in which object ball appears.

```
Select videoscene
From all
Where appear(ball)
Order by time;
```

In this case if a ball object is found in the database it will answer directly to user, but if not get it will calculate the semantic distance of ball. More semantic values will be select and the system check again in the database for query result.

4.11 Annotated Video Database

There are different ways to design the video database model. The database model layout is an important part of an information system. In order to design a good database model first we analyze user requirement, data objects and data definitions before creating the schema. The data model defines how the data is to be stored. Video is structured into scenes, shots and key frames using hierarchal data model. Several scenes may be divided into several shots where each shot has a representative key frame interest. The data model consists of parts and sequences in text representation. Each video scene is defined by start and end time. Video scenes can be defined repetitively to label as many shots as needed. Annotation attributes are defined in free text, in the sense that user can define the attributes name description, duration and high level video description. The generic data model within RDBMS is implemented by defining attributes like name of object, a detailed description, and duration with their corresponding data type. This logical level representation can be map to any physical level database. The description of video database structure basis on relational modeling, the data model basis on the annotated video and the annotated video features are stored manually in the database. The proposed system built on

client serve-server architecture. The video database structure is presented in the Figure 4.6 as follows.

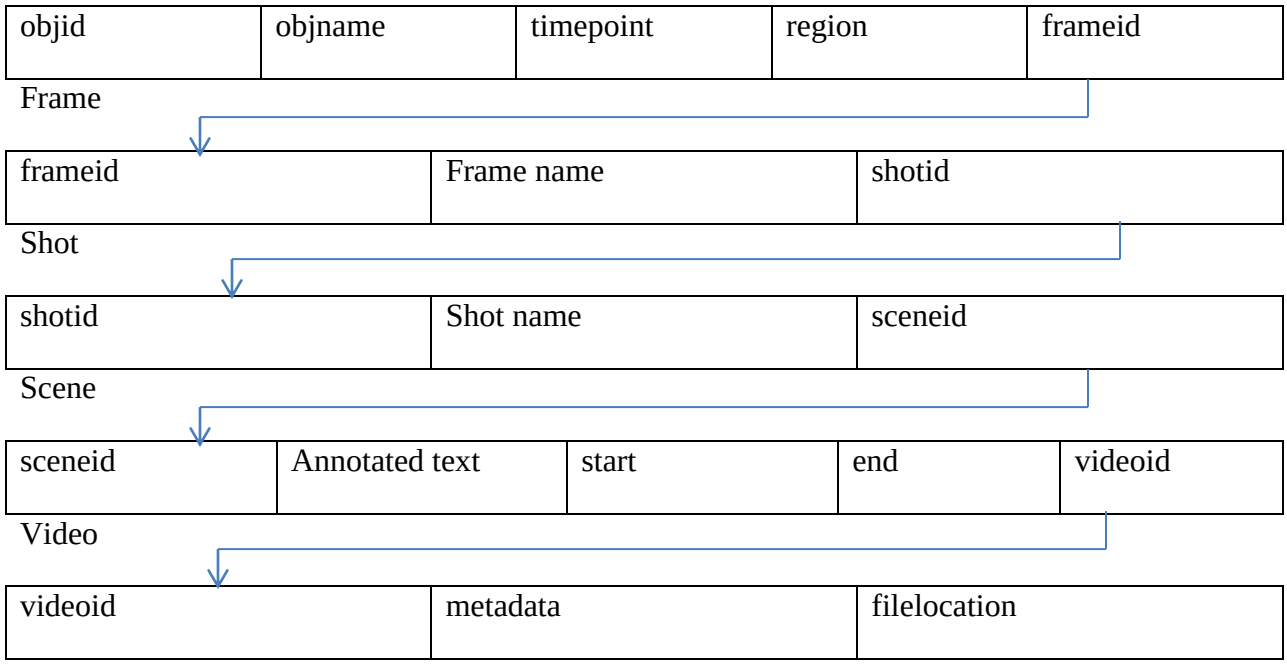


Figure 4.6: Video database structure

The annotated video XML structure is look like the below description

```
<Video Id>
    <Scene Id1>
       <Annotated Text>
          <shot1>
            <Annotated Text>
                  <Frame>
                      <Annotated text>
                              <obj>
                                  <Annotated object1>
                                  <Annotated object2>
                                   <Annotated object2>
                                 </obj>
                       </frame>
                 </shot1>
```

```
        </scene Id1>
</Video Id>
```

From the above video structure we can create a mapping of video database structure in Tables 4.1 – 4.5.

Table 4.1: Video Structure

Vid	Meta Data	FileLoc
Vid1	barcelona game	path
Vid2	Soccer Moves Tricks And Skills	path

Table 4.2: Scene structure

SceneId	Annotated Text	Duration		vid
		Start	end	
Sc1	Ball in Barcelona field	0	58	Vid1
Sc2	Training skills	23	45	Vid2

Table 4.3: Shot Structure

shotid	shotname	sceneid
S1	Goal inside bar	Sc1
S1	messi standing	Sc1
S2	Training person	Sc2

Table 4.4: Frame Structure

frameid	framename	shotid
f1	messi and fabregas	S1
f2	Messi standing	S1
f3	person with ball	S2

Table 4.5: Object Structure

objectid	objname	timepoint	region	frameid
Obj1	ball	31	polygon	f1
Obj2	messi	40	polygon	f1
Obj3	person	50	polygon	f2

4.12 Video Scene Output

This video retrieval system tries to answer user queries based on the match in annotated video database. In this process, a set of videos are returned for a query and the results are presented with one or multiple video scene and with some metadata descriptions, such as its title, and general description.

Chapter 5: Natural Language based Query Formulation for Soccer Video Retrieval

In this thesis, we have developed a video retrieval system called NLQVR (Natural Language based Query Formulation for Video Retrieval). We have chosen the SQL server management studio for storing objet entities. In our prototype system we have used Neatbeans Ide environment using the JDBC interface for the development of the design model and opencv tool for taking coordinate points.

NLQVR offers an interface that allows the storage for entities and retrieval of a video units with multi-criteria query formulation support. In the sections that follow the case studies of video retrieval prototype system are presented.

5.1 Video Database

The video data model and the video repository models proposed in this thesis are generic and can be used in different domain. However, to demonstrate the practicality of our system, we use a specific application domain such as the soccer application domain. For the sake of simplicity we limit NLQVR to manage soccer video data only. The proposed video data repository models defined at a video and its constituent units can also be used in any DBMS environment. A sample database of a soccer application domain, which we used in the prototype system, is presented below.

The tables for the chosen application domain are organized as in the following manner. The tables defined below are associated to the components of the proposed video data repository models defined at a video and its constituent units.

- **Object** (**ObjId,** name, point,regions,frmeid), information related to object such as player name, field which is uniquely identified by the object Id (Obj_Id) is captured.
- **Frmae** (**frame_Id**, name, shotid), information related to a frame which is uniquely identified by frame Id (frame_Id) is captured. This table can be conveniently associated to the video shot table of the proposed video repository models.
- **Shot** (**shot_Id**, shot_Description,sceneid), information related to the shot content of a video which is uniquely identified by shot Id (shot_Id) is captured. This table holds textual data, which describes the shot contents of of a video. This table can be

conveniently associated to the video scene table of the proposed video data repository models.

- **Scene** (**scene_Id**, annotation, start, end, videoId), the table contains information related to the video scene which is uniquely identified by scene Id (scene_Id) is captured. This table holds all complete story units of a video segment, mainly the scene annotation of a video. This table can be conveniently associated to the direct video table of the proposed video data repository models.
- **Video** (**video_Id**, Metadata, location), this table holds information related to the whole video content which is uniquely identified by video Id (video_Id) is captured. This table contains metadata information, which describes video length, and year of release of a video. This table can be conveniently associated to the video itself and it is independent of the content of a video.

5.2 Query Interfaces

Our prototype system supports two types of queries: Query by exact match and query by similarity (semantic) metrics. The query by exact match interface is mainly used to formulate syntax based on the presence of stated objects and operators in user query as a query key.

The query processing passes different stage of preprocessing components to come up to the query formulation process. Finally, the query is formulated and ready with execute button to enable a user to query video scenes from the video repository. The query by semantic interface allows searching by domain specific key word or vocabulary in the WordNet database if the threshold value is met. For the sake of simplicity, in the proposed system we have limited our operator on spatial, temporal and predicate operators which occur frequently in a football match. We presented the two main query interfaces of NLQVR as follows.

5.2.1 Query Based on Stated User Questions

In this section we elaborate different queries when user stated objects are match with the stored database entities and operators. In our prototype system, object entities are extracted from the representative key-frames from the selected images. Since key frames are still images which contains set of objects. The extracted object coordinate points are stored in the object table from the video repository. Each representative key-frame is included in video shot table in the

repository set with similar key-frames from the extracted video shots. Above all a video scene is a complete story unit where selected video shots are inserted into a repository or table in a compact representation of its content and it is stored in a set of shot ID. The video scene column contains the unique ID, annotation, start and end points and video information along with their-object based location information to represent the entire video scene or video.

Thus, any query operation deals solely with this abstraction rather than with the image itself. Object entities are thus compared based on their attribute values and the region which is located in the frame.

Different operator accepts list of object entities as argument for measuring the behavior of objects entities based on the specified syntax. The stored procedure rule computes a similarity measure for each attribute in the defined SQL query form. Each operator result values reflects how objects are matched for a given object entities. The comparison metrics for each attribute value is measured based on the coordinate points for the inclosing region objects. Based on user query the proposed system returns video scenes which have similar story lines in the video repository.

The query interface in Figure 5.1 works in the following manner, a natural language query is accepted by the system and is preprocessed to find the stated object and operators if available. The extracted object and operators are compared with those captured entities in database and operator in the stored procedure rule respectively. Finally, all relevant answers are retrieved and displayed in the appropriate graphical form. For example, 'Find videos where Messi is inside penalty area and ball inside bar'. To answer the above query the preprocessing module discards irrelevant terms which have no impact for query generations and readies to the query identification module. The extracted objects are Messi, penalty area, ball and bar and, the operator is inside respectively. The equivalent query is generated based on the stated object and operators like '`Inside @Messi,@penality area AND Inside @ball, @bar`'. Finally, the equivalent query is executed to get the result set as shown in Figure 5.1.

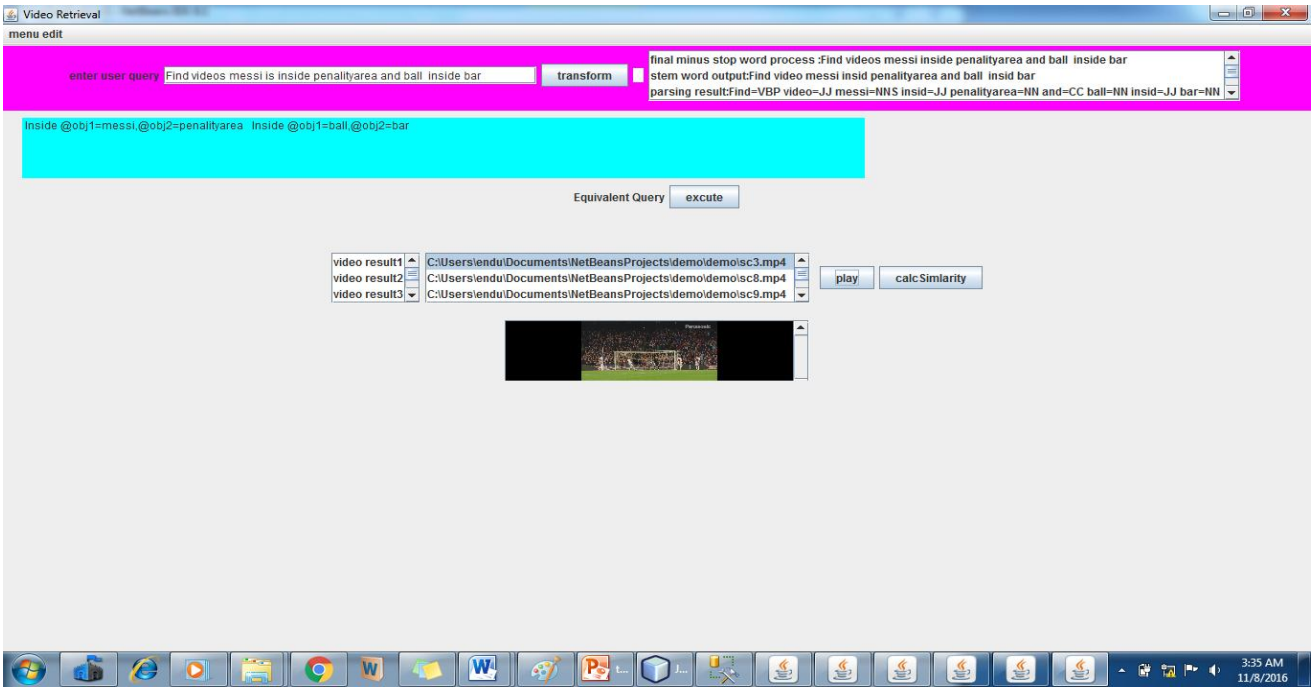


Figure 5.1: Query interface for exact stated object

5.2.2 Similarity-Based Comparison

The aim of similarity based comparison is to use knowledge domain-independent semantic concepts to get better and closer results. The main issue in semantic similarity is getting more accurate results between concepts in the user query and objects in database. In our video scene retrieval system, the similarity metrics component computes semantic terms using WordNet dictionary. Therefore, result sets are selected based on the degree similarity of object entities. We compute the semantic measures between query words appear in the user query and every object in video repository. If the stated objects meet the threshold value, then the system will retrieve videos scenes which are associated to the video data object.

The query interface shown in Figure 5.2 works in the following manner, the similarity component computes the query word and an object in video database. The degree between the query word and video object will be measured using the distance measure. The result of highest distance value is taken as key object for the query word processing. We use Wu and Palmer's distance-based similarity techniques to measure the similarity between query objects and objects in the video database [55]. The key object in the query is compared with the objects in the video database to measure their similarities. The most similar objects are selected by our conceptual

similarity method in order to be used in the query execution stage. This operation is done for every video objects which are found in the video database. For example ‘find a video scene ball inside bar’. In order to answer such query, the similarity components finds all possible synsets of the query terms and compute the distance from the object in database, in this case ‘ball’ and ‘bar’ are user interest terms and finds result sets which have meet the threshold value with the same video scene ids.

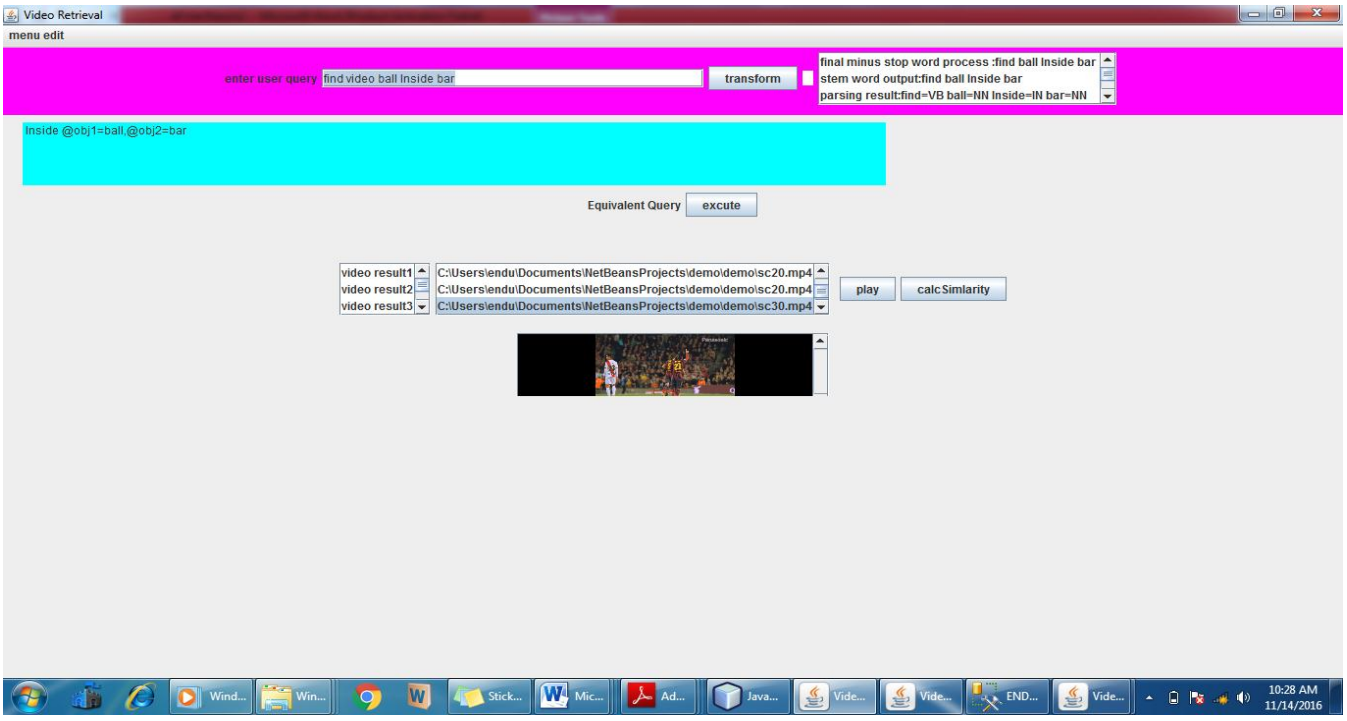


Figure 5.3: Semantic Query

Chapter 6: Implementation and Experimental Results

In this Chapter, the detailed implementation and evaluation techniques of our video query systems will be presented. First we will cover the pre-processing techniques and algorithms used in video query processing. Next, we will explain the detailed strategies and algorithms implemented in analyzing questions posed by the user so that the expected answer type, query type, and query generation will be explained. We also define different spatio-temporal operator definitions used in retrieving query from the annotated video database. Finally, the detail evaluation techniques are incorporated. Finally, the best answers are retrieved among the pool of candidate answers respectively.

6.1 Development Environment

We have used different tools and programming environments for each component implementations to deploy and execute. Java programming language with NetBeans development environment is used to work with processing natural language and to implement the algorithm developed. Java provides different libraries which make natural language processing easier, and some libraries have also been used while working with these tasks of query generation and execution. Java is a good programming tool in natural language and multimedia processing and it can be integrated with different programming environments. We have used visual studio with opencv library to capture objects and to take coordinate points which covers the object bounding. Opencv is an open source library, mainly aimed at computer vision. This library is written in C++ and its primary interface is in C++, this is used while preparing video dataset to take coordinate points of an object. We also used Microsoft SQL server 2013 for storing coordinate points in database corresponding to the object entity attributes. Since MS SQL server 2013 can support spatial database we can store different geometry data types. Basic libraries used in this work are

- Stanford-parser: is used to assign parts of speech tags to terms extracted from the user queries.
- opencv library-2.0.221: is an open source C++ library for image processing and computer vision and, used to take coordinate points from the video files.

- Ws4j-1.0.122: is used to normalize and expand concepts for different operations on WordNet.

6.2 Dataset Preparation

We have used a collection of video scene files with their annotated dataset. These videos were captured from YouTube mainly of soccer game. Currently, the database contains a total of 220 annotations, 170 object classes, 30 action annotations and 20 are semantic terms. The most frequently annotated objects in the video database are *players, ball, and field area.* We also prepared different query types since question set preparation is the main task for evaluation requirement. The queries are prepared in English language format. A state-of the- art system should pass the main criteria set along the question sets. For our system, we have collected a number of questions from spatial and temporal types. Most of the questions were spatio-temporal while the remaining is predicate type query. A sample user query is attached in Appendix A. A total of around 200 questions have been prepared to evaluate our system.

6.3 System Prototype

A prototype is developed to show the validity of the proposed video scene retrieval. Our system is capable to query all types of spatial and temporal with limited predicate query using natural language query. Thus, the system has user interfaces associated with entering user question and shows results after query executions.

6.3.1 System User Interface

User interface allows users to enter query to be transformed in appropriate processing format as shown in *Figure 6.1*. Once the query is accepted, the pre-processing module starts to remove stop words. Then a set of tokens goes to a parsing module in order to get a set of POS tags. The result of parsing is a set of nouns, verbs and adjective tags. Each set of POS are further processed to detect real objects and verbs with object detection module. The result of detection is further processed to get spatial and temporal operators. Initially the video is split into scene level based on the concept classification of the video, which is kept in repository and we store the video object coordinate points in SQL server database. Initially we have not use any algorithm to check the scene boundary detection rather we use manually based on the concept demarcation. To demonstrate the system, we have used different soccer videos with different and similar scene.

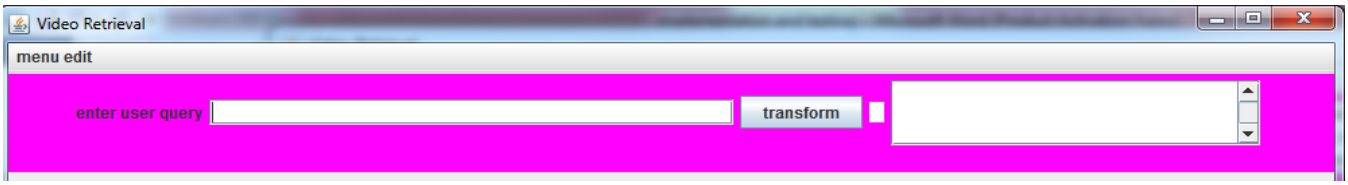


Figure 6.1: User query input

6.3.2 Natural Language Preprocessing and Detection of Entities

NLP helps for further processing user queries to make questions ready for query execution phase. Questions which are accepted from the system needs extra efforts for processing before the subsequent query subsystems act on them. Unless user query are preprocessed to a uniform way the query retrieval will be affected. Therefore we have further broken down preprocessing such as stop word removal and stemming. From the user query side, the questions are checked using stop word removal component for removing irrelevant words and, for the implementation purpose we identify words as question tags, question marks etc., We then took the result into stemmer component, which helps for removing the morphological inflexional endings words. For better query formulation stemming has good role since we store object name annotation in database. Retrieving in database requires exact match since, stemming helps the overhead of this problem by giving a common format. For our work we have applied stemming and didn't consider morphological variation searching.

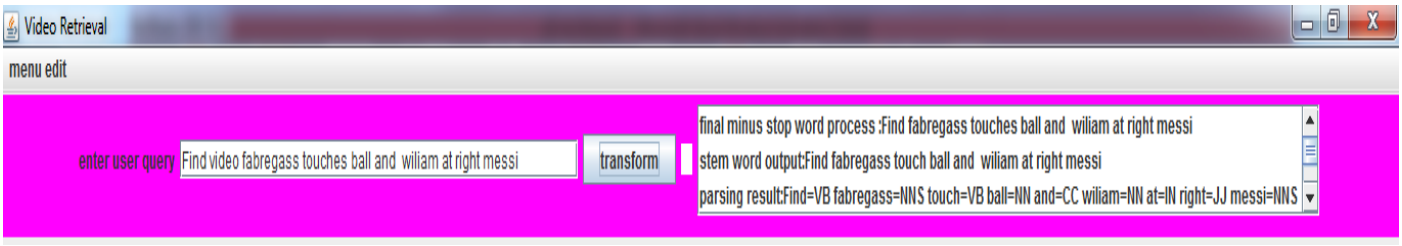


Figure 6.2: Pre-processing

A set of tokens are generated from the stemmed word output as shown in Figure 6.2 and, for further identification of relevant query terms we process the object and event module. In Figure 6.3 shown candidate nouns and verbs are listed based on the output result of parsing after executing the parsing component module. List of noun contains words tagged as nouns with their type and list of verbs are tagged as verbs respectively. Nouns are further processed to be taken as objects and verbs are taken to be events ones they are normalized.

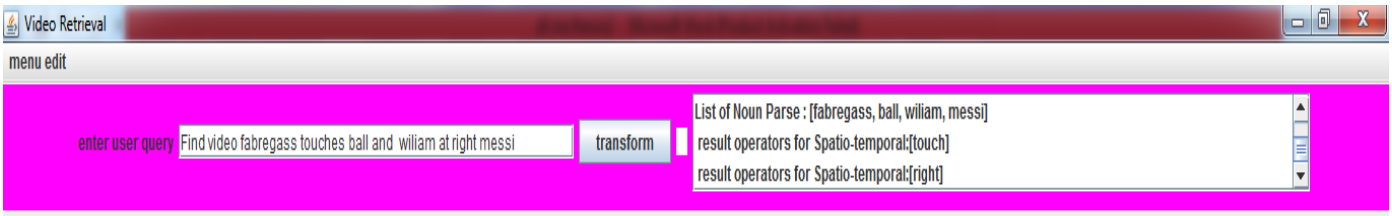
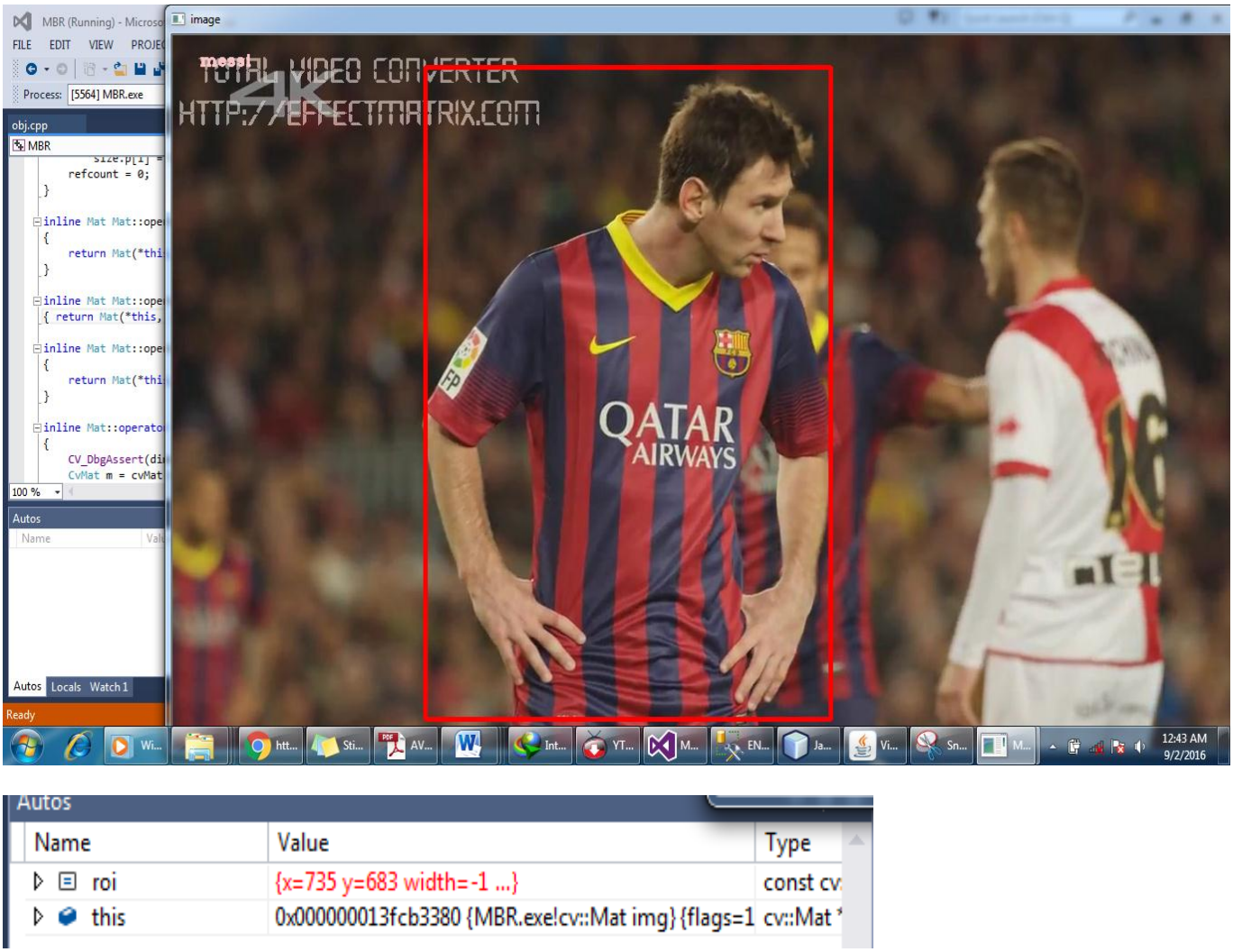


Figure 6.3: Objects and event detection

Here the detected object and events are refined further for operator detection. The candidate operators are analyzed to determine the category of queries to which it belongs, what will be the expected number of entity type, and the argument focus were done. The detected operators will determine the question types, the question focuses (if present) and the expected answer types. Operator identification involves techniques of identifying the question particles which helps in stating what the question is about. To do so, we have identified object type which goes to the operators. Each operators are defined in the stored procedure programing to identify which object parameters goes inline to be execute and what query is possess. We have defined the detailed investigation to find out what specific operators are very important in telling what types of questions are sought so far. The expected query return type is directly related to the operator type and the arguments it takes. For example, for the query "show a video where ball is inside bar" the object "ball" and "bar" are detected from the object detection module. Inside is a spatial operator which we get from the operator refinement module. Since 'inside' operator takes two object argument and will create a valid format like inside(ball,bar). In our case we will generate equivalent spatial or temporal query forms which are taken to be later executed. Objects in user query were cross checked so that unrelated objects or misspelled entities are removed and will give suggestions based on the available corresponding entities in the database. We also give a chance to perform similarity metrics for relatedness entities based on the user interest.

6.3.3 Taking Object Points (MBR)

The minimal enclosing rectangle in opencv tools provides a method of filtering coordinate points. In our case we use MBR to approximately estimate the object coordinate points. We have taken four coordinates points corresponding to the (x, y) in space with pixel value. MBR enclosing objects contains spatially the n-dimensional data object and a pointer to the file containing the actual representation of the object.



Autos

Name	Value	Type
roi	{x=735 y=683 width=-1 ...}	const cv
this	0x000000013fcb3380 {MBR.exe!cv::Mat img} {flags=1	cv::Mat *

Figure 6.4: Extracting coordinate MBR points

6.3.4 Query Processing and Generation

Our query processing module allows formulating equivalent query based on the object and operators stated in user query as shown in the Figure 6.5. The query generation works based on the available spatio-temporal query operator and with some predicate operators. Questions that are not spatio-temporal type and predicate operators will not be processed fully as it is beyond the scope of this thesis work, in our implementation we have included a set of specific predicate operator like kick, pass, from, score etc. The spatio-temporal query will be further classified to different question types based on the operator type and arguments. This classification will help in locating exactly the correct answer for later stage of video query process. Once we have identified the operator, object and argument number we can generate equivalent query based on the syntax definitions. Our operator definition holds all mathematical rules to compute spatial

and temporal query types. We have used stored procedure programming to define spatial and temporal operators. Each operator takes argument based on their definition and, it will operate on individual object to check whether it is true or not. The query generated result works based on the available operators and the return types respectively and, later will be passed to the query processing component. For example, when user enters a query "Find video Fabregass touches ball and Wiliam at right Messi" the equivalent query will be generated after performing the preprocessing steps as described in above scenario and, it generate equivalent query which is directly executed in the database servers.

Equivalent query="`Touch @obj1=fabregass,@obj=ball AND Right @obj1=wiliam,@obj2=messi`".

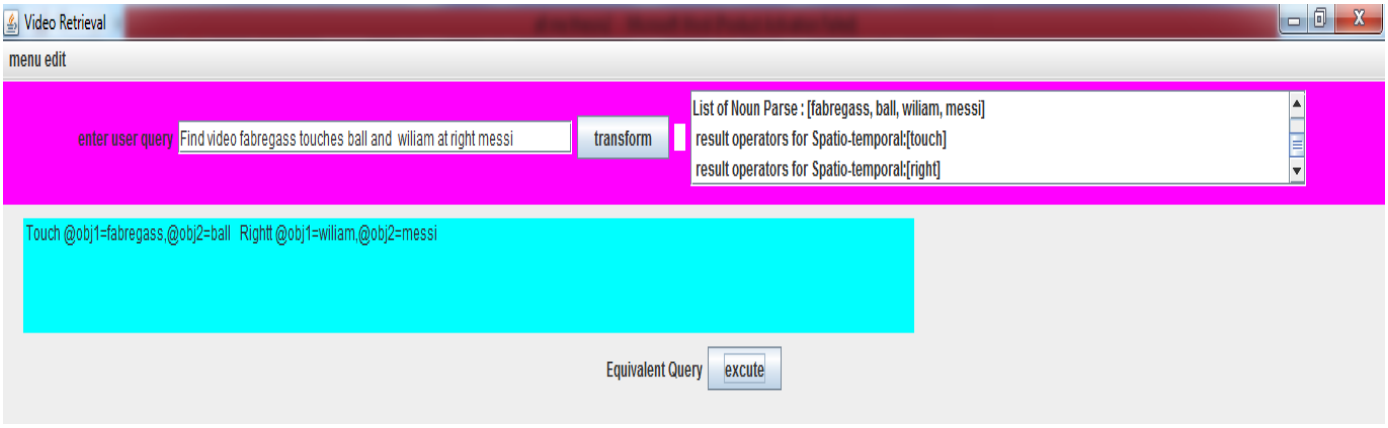


Figure 6.5: Equivalent Query Generation

6.3.5 Query Retrieval and Result

After generating equivalent query format in the above sections the final query will be loads to the SQL query processing for last result. The query processing will find all possible solution from the video database and, the final query result is a collection of video scenes which have similar story background videos. Having exact match queries, we also store similar object names in the database. The similarity metrics is done to find similar concepts which are specified in different naming. For this, object and event detection from the user query will be checked if there exist similar objects and event concepts in the video database; if they are similar, they will be set as output result in the output screen window. For example, the object name "goal" and "goalkeeper" from the user query have a similarity value *sim('*goal*', '* goalkeeper*') = 0.21*, which in our case is taken to be not similar as we took a similarity threshold of 0.43. There is no

standard similarity threshold as per the knowledge of the researcher but we have checked relatedness between stored terms and we have found 0.43 as an average value. After setting the threshold value the objects which qualify the measure value are retrieved based on the description of scene level and, unrelated objects are removed from the query processing pipeline set.

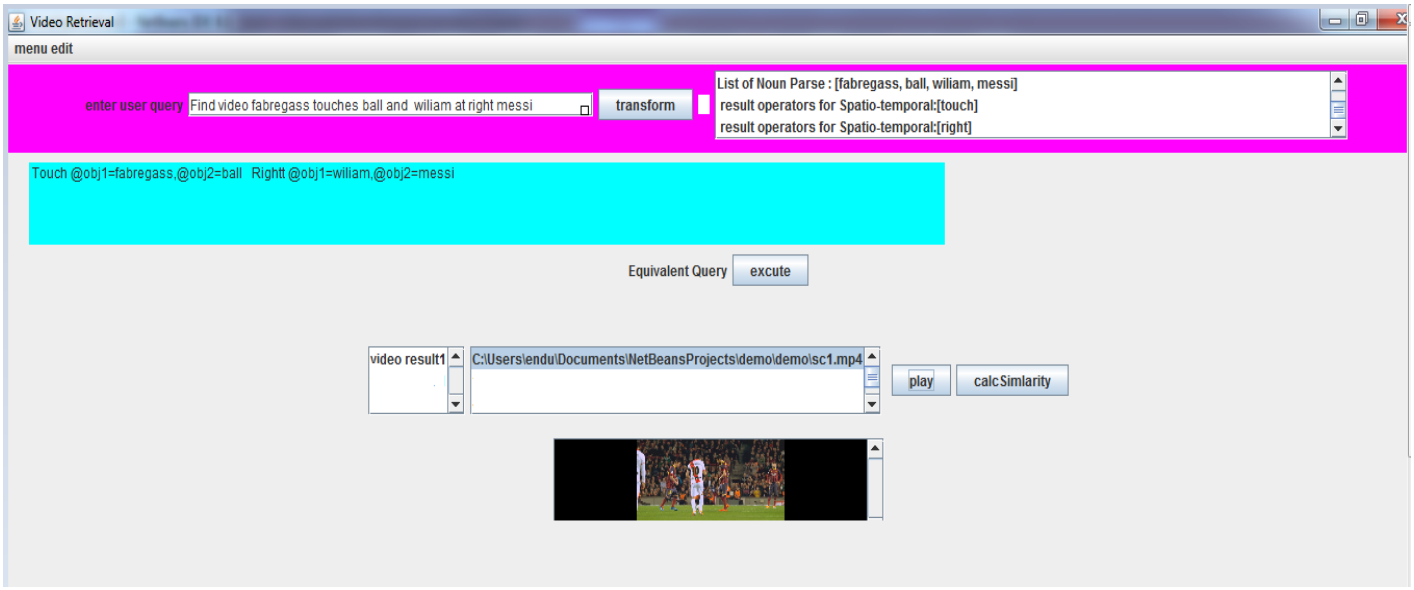


Figure 6.6: Sample query result

6.4 Evaluation

We have developed a prototype of the video retrieval system based on the approach described in Chapter 4. The data used in the experiments consist of total number 155 queries from soccer video. Some of these queries are object parameters which are stated objects in the database. In other words, they would correspond to exact matches. The 83% of the query words did not appear in the database. The developed prototype not only returns exact matches but also returns video scenes that could match semantically with the query word. The system was implemented on standalone desktop applications, where user queries are issued on query interface and the retrieval is performed on the server side. The program was implemented on client server architecture. For evaluations we use spatio-temporal representations for different types of actions, such as inside, touch, overlap, meet and directional relations like south, north, south, east and temporal operator like before. We also define domain-specific queries such as “midfield”, “penalty area”, etc. Figure 6.6 shows an example of query retrieval for the questions ‘Find video scene where Fabregass touches ball and Wiliam at right Messi. We evaluate the full system

based on natural language query which relies on the spatial and temporal operators with object as parameter process. The proposed system answers queries based on object of interest using different spatio-temporal operators. The performance of our system is believed to be evaluated based on the result query. The evaluation includes, how much of the queries are correctly answered, how much of the queries receive wrong answer, and No answer as well as whether the retrieved queries qualify an answer or not.

6.4.1 Evaluation Criteria

The evaluation criteria mainly focus on the accuracy of the answers returned. Our system was evaluated as the percentages of the queries were perfectly answered by our system. We execute the prepared queries to see which video scenes would be retrieved by the system. For each query, the returned answer set was compared with the correct answers from the expected answer sets. When we perform the evaluation firstly we give the video files to the evaluator and while doing this the expected answers are determined based on the watched videos. Finally, users formulates query and evaluate the system and, the system result is compared to the expected result set.

The accuracy was measured by using the known metrics. Evaluation of the system accuracy (precision) is calculated as the number of correctly answered queries over the total list of answers (correct, wrong, and No Answer). The recall is also calculated as number of correctly answered questions among the list of expected answer sets where queries will be first analyzed for the presence of correct answers. Percentage computation is done for correct answers, wrong answers, and No answers over the total answers which is the main evaluation criteria for many query systems in addition to precision and recall. The formula is as follows.

$$Precision = \frac{Correct\ answer}{correct\ answers + wrong\ answers + No\ answers}$$

$$Recall = \frac{correct\ answers}{Correct\ answers + missed\ answers}$$

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

6.4.2 Query Retrieval Evaluation

The video query retrieval system has been evaluated based on the presence of correct answers from the set of queries. From the total questions 121 gives correct answer and, 18 gives wrong answer and the rest 16 gives null result. Using the above metrics we got the following results.

Precision=0.78

Recall=0.87

F-measure= 0.82

Here the evaluation is done based on the number of expected answer set. Besides, this evaluation we considers all the query results are believed to have the correct answer for a user query. The results indicate that 78% of the results in the answer set are correct, and 87% of all possible correct answers appear in the answer set. These are the results with respect to the best F-measure value (0.82). Regarding the result, the object can have multiple senses and we will do similarity metrics. The similarity algorithm calculates the query word which is related to a video object, if a sense of the query word is similar to a sense of the video object; the query term treated as a sense of the video object for query result. In this case, all terms will be treated as similar words, and the video scene containing that video object will be selected into the answer set. For example, the word "ball" is related with the sense of the word 'round shapes', When the video frames containing a spherical shapes are searched, the video frame containing a ball senses will be set an answer. In general the above experiment can be enhanced when the parsing module identifies each POS to each category word and, correct annotation dramatically improves the result when it describes the scene cut points properly.

6.4.3 User Evaluation

We also evaluate the proposed system from the user's perspective. We have evaluated the performance of our system with the query that has been given for different peoples. There were 20 peoples which have selected randomly to test the system. 13 of them are postgraduate students at Addis Ababa University who have basic knowledge on video related concepts, and the rest are individuals from different profession. The selection is based on the assumption that those with video related knowledge can see and evaluate technical issues such as s scene concepts and natural language processing, while others may evaluate the applicability, accuracy and importance of the system.

The evaluation matrix is prepared and given to them in the form of questionnaire as shown in *Table 6.2*, in which they put the weight of each criterion according to their view. Weights indicate that how good the query result is correct, where 1 indicates lower value and 5 is the higher weight. User's evaluation is recorded as shown in Table 6.2 and analyzed in detail. Rows of *Table 6.2* show evaluation values of each individual and columns show evaluation criteria listed on the questionnaire as shown in *Table 6.1*.

Table 6.1: Questionnaire

NO	Questions
1.	The result videos has the same scene structure (story)
2.	Performance measure
3.	Object stated in the user query are found in the video result.
4.	User interface
5.	All the operators' definition in the user query is found in the videos.
6.	usability

1 low 2 fair, 3 good 4 very good 5 excellent

Table 6.2: User evaluation

	Q1	Q2	Q3	Q4	Q5	Q6
1	4	4	4	2	5	5
2	5	4	3	3	4	5
3	4	4	4	1	5	4
4	3	5	4	2	4	4
5	4	4	3	3	3	4
6	4	4	3	4	2	3
7	4	5	3	2	4	2
8	3	4	2	3	4	3

9	3	4	3	2	4	3
10	4	4	2	3	2	2
11	4	4	4	4	3	4
12	3	5	5	3	3	4
13	2	4	3	2	3	5
14	2	5	2	4	4	3
15	4	4	2	3	4	4
16	4	4	3	2	5	5
17	4	5	3	2	4	5
18	4	4	2	3	4	5
19	3	4	4	3	2	4
20	4	2	4	2	5	3
Average	4	3	4	2	5	4

With Cr1, the evaluation result shows the story units of video scenes are correctly fetched in the video database. This good query result indicates that the qualities of the video annotations are properly tagged as a result of proper object and event identification process.

The performance of the system shown in Cr2 has been rated to be good. The result shown in Cr3, where object stated in user query found in video result has been rated to be good as all the objects in the annotation are found in the video files. Cr4 tests how easy or difficult the UI; the average rated result is fair. The implemented system provides most of the operator definition as a result set so that, the result operator's results are found in the video scenes as shown in evaluation result of Cr5. Finally, the result of usability testing provides the opportunity to evaluate the system by studying how real users actually use the system. From the rated result data we find out the UI for a system fits with users' needs and expectations.

Chapter 7: Conclusion and Future Works

7.1 Conclusion

Video is a collection of related concepts represented in a sequential set of image frames; it needs a great deal of concept dependency processing to give a semantic description on the video content. There are a lot of works has been done in the area of natural language based video queries; almost all works retrieve videos on frame based level. Results from such queries are inadequate in describing the contents of a video entirely.

Most casual users or non-experts in database are highly dependent on using natural language queries. These NLP queries have a lot of advantages that are used to fire query without knowing the schema structure and SQL syntax. NLP queries are just a set of words and handling these terms to formulate query is a very difficult task.

In this study, we presented natural language based video retrieval. In order to implement the natural language interface, we used preprocessing modules like stop word removal and stemming word. We also used Stanford parsing algorithm to identify the possible associated POS terms. Each POS term is further processed using entity detection algorithm. The extracted elements are objects, events, and activities. The entity refinement algorithm is used to find the final candidate object and spatio-temporal operators which help for query formulation stage and spatio-temporal relations. We define spatio-temporal and predicate based video retrieval data model. The model presented here identifies coordinate points of objects using rectangular areas or regions. So, it is possible to compute spatial relationships between two regions using the coordinate points since the identified object properties are covered by rectangles.

Moreover, the proposed study implemented semantic search when exact match not found from the database. Using WordNet the system retrieves the most similar objects to the words given in the user query.

Appropriate tools and techniques have been identified to develop a prototype system for the research for the soccer videos. The proposed system has been tested with different Spatio-temporal queries and it has been found to be accurate 78% of the time.

7.2 Contributions

The contributions of this research work are:

- Generate high level language query form for NLP query
- Handles complex query
- scene level video query
- entity detection algorithm
- Implemented entity Refinement algorithm
- Implemented query formulation representation

7.3 Future works

This research work explores different areas that can be further improved as well as some components that should be implemented and integrated for better functioning of the system. Some of the future works include:

- Query optimization
- Implementing and identifying object regions
- Implementing trajectory or moving object queries

References

[1] Mohand Hacid, Cyril Decleir, and Jacques Kouloumdjian. "A database approach for modeling and querying video data." *IEEE transactions on Knowledge and Data Engineering* 12, no. 5 (2000): 729-750.

[2] Xavier Carreras and Lluís Màrquez. "Introduction to the CoNLL-2005 shared task: Semantic role labeling." In *Proceedings of the Ninth Conference on Computational Natural Language Learning*, pp. 152-164. Association for Computational Linguistics, 2005.

[3] Mehmet Emin Dönderler, Ediz Şaykol, Umut Arslan, Özgür Ulusoy, and Uğur Güdükbay. "BilVideo: Design and implementation of a video database management system." *Multimedia Tools and Applications* 27, no. 1 (2005): 79-104.

[4] Onur Kucuktunc, Ugur Gudukbay, and Ozgur Ulusoy. "A Natural Language Based Interface for Query Specification in a Video Database Management System." *MultiMedia, IEEE* 14, no. 1 (2007): 83-89.

[5] Dessalegn Mequanint, "similarity-based video retrieval: modeling and processing", Unpublished Masters Thesis, Department of Computer Science, Addis Ababa University, 2004.

[6] Guzen Erozel, Nihan Cicekli, and Ilyas Cicekli. "Natural language querying for video databases." *Information Sciences* 178, no. 12 (2008): 2534-2552.

[7] Mesru Koprulu, Nihan Cicekli, and Adnan Yazici. "Spatio-temporal querying in video databases." *Information Sciences* 160, no. 1 (2004): 131-152.

[8] Onur Kucuktunc, Ugur Gudukbay, and Ozgur Ulusoy. "A Natural Language Based Interface for Query Specification in a Video Database Management System." *MultiMedia, IEEE* 14, no. 1 (2007): 83-89.

[9] Boris Katz, Jimmy Lin, Chris Stauffer, and Eric Grimson. "Answering Questions about Moving Objects in Surveillance Videos." In *New directions in question answering*, pp. 145-152. 2003.

[10] Majdi Owda, Zuhair Bandar, and Keeley Crockett. "Conversation-based natural language interface to relational databases." In *Web Intelligence and Intelligent Agent Technology Workshops, 2007 IEEE/WIC/ACM International Conferences on*, pp. 363-367. IEEE, 2007.

[11] Walid Aref, Moustafa Hammad, Ann Catlin, Ihab Ilyas, Thanaa Ghanem, Ahmed Elmagarmid, and Mirette Marzouk. "Video query processing in the VDBMS testbed for

video database research." In*Proceedings of the 1st ACM international workshop on Multimedia databases*, pp. 25-32. ACM, 2003.

[12] Akshay Satav, Archana Ausekar, Radhika Bihani, and A. Shaikh. "A Proposed Natural Language Query Processing System." *International Journal of Science and Applied Information Technology* 3, no. 2 (2014).

[13] Ashish Kumar and Kunwar Singh. "Natural Language Interface to Databases: Development Techniques." *Elixir International Journal* (2013).

[14] B. V Patel and B. B. Meshram. "Content based video retrieval systems."*arXiv preprint arXiv:1205.1641* (2012).

[15] Alessandra Giordani, and Alessandro Moschitti. "Translating Questions to SQL Queries with Generative Parsers Discriminatively Reranked." In*COLING (Posters)*, pp. 401-410. 2012.

[16] Tomoki Masuda, Daisuke Yamamoto, Shigeki Ohira, and Katashi Nagao. "Video scene retrieval using online video annotation." In *Annual Conference of the Japanese Society for Artificial Intelligence*, pp. 54-62. Springer Berlin Heidelberg, 2007.

[17] Vani Jain and Ramazan Aygün. "Spatio-temporal querying of video content using SQL for quantizable video databases." Journal of multimedia4, no. 4 (2009): 215-227.

[18] L. Chen II, M. Tamer Özsu, V. Oria: "Modeling Video Data for Content Based Queries: Extending the DISIMA Image Data Model", MMM 2003: 169-189.

[19] B. Sujatha, Dr S. Vishwanadha Raju, and Humera Shaziya. "A Survey of Natural Language Interface to Database Management System." *International Journal of science and Advance Technology* 2, no. 6 (2012): 56-60.

[20] Christopher Manning. "Generating typed dependency parses from phrase structure parses." (2008).

[21] Charles Sutton and Andrew McCallum. "Joint parsing and semantic role labeling." In *Proceedings of the Ninth Conference on Computational Natural Language Learning*, pp. 225-228. Association for Computational Linguistics, 2005.

[22] Heeyoung Lee, Marta Recasens, Angel Chang, Mihai Surdeanu, and Dan Jurafsky. "Joint entity and event coreference resolution across documents." In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 489-500. Association for Computational Linguistics, 2012.

[23] Arpit Sharma, Nguyen VO, Somak Aditya, and Chitta Baral. "Identifying Various Kinds of Event Mentions in K-Parser Output." In *Proceedings of the 3rd Workshop on EVENTS at the NAACL-HLT*, pp. 82-88. 2015.

[24] William Phillips and Ellen Riloff. "Exploiting role-identifying nouns and expressions for information extraction." In *Proceedings of International Conference on Recent Advances in Natural Language Processing*, pp. 165-172. 2007.

[25] D. Zhong and S-F. Chang. "Video object model and segmentation for content-based video indexing." In *Circuits and Systems, 1997. ISCAS'97., Proceedings of 1997 IEEE International Symposium on*, vol. 2, pp. 1492-1495. IEEE, 1997.

[26] Neelu Nihalani, Sanjay Silakari, and Mahesh Motwani. "Natural language interface for database: a brief review." (2011).

[27] Natalia Silveira, Timothy Dozat, Marie-Catherine Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Christopher Manning. "A Gold Standard Dependency Corpus for English." In *LREC*, pp. 2897-2904. 2014.

[28] Jürgen Assfalg, Marco Bertini, Carlo Colombo, Alberto Bimbo, and Walter Nunziati. "Semantic annotation of soccer videos: automatic highlights identification." *Computer Vision and Image Understanding* 92, no. 2 (2003): 285-305.

[29] Robert Sorschag. "A high-level survey of video annotation and retrieval systems." *International Journal of Multimedia Technology* 2, no. 3 (2012): 62-71.

[30] Yan Liu and Fei Li. "Semantic extraction and semantics-based annotation and retrieval for video databases." *Multimedia Tools and Applications* 17, no. 1 (2002): 5-20.

[31] Lawrence Rowe, John Boreczky, and Charles Eads. "Indexes for user access to large video databases." In *IS&T/SPIE 1994 International Symposium on Electronic Imaging: Science and Technology*, pp. 150-161. International Society for Optics and Photonics, 1994.

[32] Shih-Fu Chang, William Chen, Horace Meng, Hari Sundaram, and Di Zhong. "VideoQ: an automated content based video search system using visual cues." In *Proceedings of the fifth ACM international conference on Multimedia*, pp. 313-324. ACM, 1997.

[33] Tony Kuo, and Arbee Chen. "Content-based query processing for video databases." *IEEE Transactions on Multimedia* 2, no. 1 (2000): 1-13.

[34] Frank Meng, and Wesley Chu. "Database query formation from natural language using semantic modeling and statistical keyword meaning disambiguation." *Computer Science Department. University of California*(1999).

[35] Lawrence Rowe, John Boreczky, and Charles Eads. "Indexes for user access to large video databases." In *IS&T/SPIE 1994 International Symposium on Electronic Imaging: Science and Technology*, pp. 150-161. International Society for Optics and Photonics, 1994.

[36] Khushboo Khurana and M. B. Chandak. "Study of Various Video Annotation Techniques." *International Journal of Advanced Research in Computer and Communication Engineering* 2, no. 1 (2013): 909-914.

[37] Jenny Yuen, Bryan Russell, Ce Liu, and Antonio Torralba. "Labelme video: Building a video database with human annotations." In *2009 IEEE 12th International Conference on Computer Vision*, pp. 1451-1458. IEEE, 2009.

[38] Hyowon Lee, Alan Smeaton, and Jonathan Furner. "User-interface issues for browsing digital video." (1999).

[39] Tomo Sjekavica, Ines Obradović, and Gordan Gledec. "Semantic Annotation and Retrieval using Multimedia Ontologies." *International journal of computers and communications* 8 (2014): 140-148.

[40] Ilaria Bartolini, Marco Patella, and Corrado Romani. "SHIATSU: tagging and retrieving videos without worries." *Multimedia tools and applications* 63, no. 2 (2013): 357-385.

[41] Valentina Malaxa and Ian Douglas. "A framework for metadata creation tools." *Interdisciplinary Journal of Knowledge and Learning Objects* 1 (2005): 151-162.

[42] Ruud Bolle, B-L. Yeo and Minerva Yeung. "Video query: Research directions." *IBM Journal of Research and Development* 42, no. 2 (1998): 233-252.

[43] YouTube, www.youtube.com

[44] Metacrawler, www.metacrawler.com.

[45] Informedia,Carnegie Mellon University,Available from<http://www.informedia.cs.cmu.edu

[46] Hyowon Lee, Alan Smeaton, and Jonathan Furner. "User-interface issues for browsing digital video." (1999).

[47] Miller George. "WordNet: a lexical database for English." Communications of the ACM 38, no. 11 (1995): 39-41.

[48] Fotis Kokkoras, Haitao Jiang, I. Vlahavas, Ahmed Elmagarmid, Elias Houstis, and Walid Aref. "Smart VideoText: a video data model based on conceptual graphs." *Multimedia Systems* 8, no. 4 (2002): 328-338.

[49] Kutluhan Erol and V. S. Subrahmanian. "Advanced Video Information System: Data Structures and Query Processing."

[50] Ion Androutsopoulos, Graeme Ritchie, and Peter Thanisch. "Natural language interfaces to databases–an introduction." Natural language engineering 1, no. 01 (1995): 29-81.

[51] William Woods, Ronald Kaplan, and Bonnie Nash-Webber. The lunar sciences natural language information system: Final report. Bolt, Beranek and Newman, Incorporated, 1972.

[52] Michael Bendersky, Donald Metzler, and Bruce Croft. "Effective query formulation with multiple information sources." In Proceedings of the fifth ACM international conference on Web search and data mining, pp. 443-452. ACM, 2012.

[53] Tang Xuri. "English Morphological Analysis with Machine-learned Rules." (2006).

[54] Lingling Meng, Runqing Huang, and Junzhong Gu. "A review of semantic similarity measures in wordnet." International Journal of Hybrid Information Technology 6, no. 1 (2013): 1-12.

[55] Zhibiao Wu, and Martha Palmer. "Verbs semantics and lexical selection." In Proceedings of the 32nd annual meeting on Association for Computational Linguistics, pp. 133-138. Association for Computational Linguistics, 1994.

Appendixes

Appendix A: Sample Queries

List of selected sample queries

no	Queries	Correct answer	Wrong answer	No answer
1	find video ball is inside bar			
2	find a video where inesta meets with messi			
3	find a video where ball insidee bar and messi meet with ball			
4	Find video ball in bar			
5	Find video messi appear at right of wiliam			
6	Find video fabregass touches ball			
7	Find video messi overlaps with neymar			
8	Find video messi and ball appear in left			
9	Find video fabregass touches ball and messi at right of wiliam			
10	Find video ball is inside bar and messi appear left of ball			
11	Find video messi insidee penalityarea			
12	Find video fabregass inside penality area			
13	find video neymar touches ball			
14	Find video messi and neymar insidee penalityarea			
15	Find video ball appear to north of messi			
16	Find video messi appear before 3 minute			
17	Find video fabregass meets with neymar			
18	Find video fabregass appear to the left of messi			
19	Find video ball touches bar			
20	Find videos ball is insidee penalityarea			
21	Find videos goalkeeper meets ball			
22	Find videos where ball pass over the bar			

Appendix B: List of Operators

	operators	spatial	temporal	predicate	NLP
1	inside	✓			
2	equal		✓		
3	East	✓			
4	west	✓			
5	north	✓			
6	Left	✓			
7	right	✓			
8	meet	✓			
9	overlaps	✓			
10	touch	✓			
11	Pass			✓	✓
12	Throw			✓	✓
13	during		✓		
14	before		✓		
15	after		✓		
16	of				✓
17	Score				✓
18	from				✓
19	Not inside				✓
20	Not overlap				✓
21	in				✓
22	to				✓
23	Appear			✓	✓

Appendix C: Sample Code

```
public void generatesql() throws SQLException{
                String mjp[]=object.split(", ");
                for(int y=0;y<mjp.length; y++){
    String resultevent;
    String ji = listnoun.toString();
    String sp=listadj.toString();
    String ep=listverb.toString();
    System.out.println("generated objects : "+loginTokenn);
    String opobj=listnounrr.toString();
    String opsp=listadjop.toString();
    String verbAll=listverbop.toString();
    String verbAll1[]=verbAll.split(",");
      for(int end=0; end<verbAll1.length; end++){
          System.out.print(verbAll1[end]+"verb");
    }
          String All []=opsp.split(",");
    for(int end=0; end<All.length; end++){
        System.out.print(All[end]);
    }
    String opevent=listverbop.toString();
    System.out.println("generated object operators: "+opobj);
     String gop[]=opobj.split(" ");
     String oo=null;
     String oo2=null;
```

```
String object =loginTokenn.substring(1,loginTokenn.length(- 1);
        String gopobj[]=objects.split(",");
        for(int i=0; i<gopobj.length; i++){
             oo=gopobj[i];
          }
          for(int k=0;k<mjp.length; k++){
                 int ethio=0;
                 System.out.print("out"+mjp[k]+"met");
                 String Myop=mjp[k];
                 String op2=Myop;
            if(k==1 || k==3){
               ethio=ethio+2;
            }
              if(k%2==0 && k==0)  {
                 GR.append("");
          }
             else if(k%2==1){
              GR.append(" "+""+" ");
                        }
           for(int ob=ethio+1;ob<gopobj.length; ob++) {
               String ino=gopobj[ethio];
               System.err.println("input object arguments"+ino);
               System.out.println("generated object  : "+opobj);
                      obj=gopobj[ethio];
                      obj2=gopobj[ob];
if (op2.equals("inside") {
```

```
Connection connection =
DriverManager.getConnection(databaseURL,user,password);

          CallableStatement stmt=connection.prepareCall ("{? =
call Inside(?, ?)}");

      CallableStatement st=connection.prepareCall ("{call
Insidefinal(?, ?)}");

       stmt.registerOutParameter (1, Types.INTEGER);

       stmt.setString(2,obj);

       stmt.setString(3,obj2);

       st.setString(1,obj);

       st.setString(2,obj2);

       rs=stmt.executeQuery();

            if(rs.next()){

        System.out.print("value exist"+rs.getString("param1"));

        GR.append(Inside+" "+"@obj1="+obj+","+"@obj2="+obj2);

        st.executeUpdate();

   inside=true;

    }

   else{

      System.out.print("no no value");

      GR.append("the objects are not appear in inside ");

    }

       }

else if (op2.equals("left") {

Connection connection =
DriverManager.getConnection(databaseURL,user,password);

     CallableStatement stmt=connection.prepareCall ("{? = call
Leftt(?, ?)}");
```

```
     CallableStatement st=connection.prepareCall ("{call
finaleft(?, ?)}");

    stmt.registerOutParameter (1, Types.INTEGER);

    stmt.setString(2,obj);

    stmt.setString(3,obj2);

    st.setString(1,obj);

    st.setString(2,obj2);

    rs=stmt.executeQuery();

       if(rs.next()){

   System.out.print("value exist"+rs.getString("param1"));

   GR.append(Left+" "+"@obj1="+obj+","+"@obj2="+obj2);

   st.executeUpdate();

   left=true;

    }

   else{

      System.out.print("no value");

      GR.append("the objects are not appear in left ");

    }  }

    if (op2.equals("right") {

Connection connection =
DriverManager.getConnection(databaseURL,user,password);

    CallableStatement stmt=connection.prepareCall ("{? = call
Rightt(?, ?)}");

    CallableStatement st=connection.prepareCall ("{call
finalright(?, ?)}");

    stmt.registerOutParameter (1, Types.INTEGER);

    stmt.setString(2,obj);
```

```
    stmt.setString(3,obj2);
      st.setString(1,obj);
      st.setString(2,obj2);
      rs=stmt.executeQuery();
            if(rs.next()){
     System.out.print("value exist"+rs.getString("param1"));
     GR.append(Right+" "+"@obj1="+obj+","+"@obj2="+obj2+" ");
        st.executeUpdate();
   right=true;
    }
   else{
      System.out.print("no value");
      GR.append("the objects are not appear in right ");
    }
       }
else if (op2.equals("touch") {
Connection connection =
DriverManager.getConnection(databaseURL,user,password);
     cs.setInt(1, id);
     CallableStatement stmt=connection.prepareCall ("{? = call
Touch(?, ?)}");
     CallableStatement st=connection.prepareCall ("{call
finaltouch(?, ?)}");
    stmt.registerOutParameter (1, Types.INTEGER);
    stmt.setString(2,obj);
    stmt.setString(3,obj2);
    stmt.setString(3,obj2);
```

```
            st.setString(1,obj);
            st.setString(2,obj2);
            rs=stmt.executeQuery();
             if(rs.next()){
     System.out.print("value exist"+rs.getString("param1"));
     GR.append(Touch+" "+"@obj1="+obj+","+"@obj2="+obj2);
       st.executeUpdate();
   touch=true;
             }
   else
    {
      System.out.print("no value");
      GR.append("the objects are not appear in touch ");
    }
         }
  }
}
// stored Procedure syntax for Inside operator
USE [videoS]
GO
/****** Object:  StoredProcedure [dbo].[Inside]  ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create procedure [dbo].[Inside]
(
@obj1  varchar(150),
```

```
@obj2  varchar(150)
)
as
SELECT
object.frameid,frame.shotid,shot.sceneid,scene.scpath,objid,obje
ct.minx, object.miny, object.maxx,object.maxy  into inside1
FROM [dbo].[object],[dbo].[frame],[dbo].[shot],[dbo].[scene]
where  object.frameid=frame.frameid and frame.shotid=shot.shotid
and shot.sceneid=scene.sceneid and objname=@obj1
SELECT
object.frameid,frame.shotid,shot.sceneid,scene.scpath,objid,obje
ct.minx, object.miny, object.maxx,object.maxy  into inside2
  FROM [dbo].[object],[dbo].[frame],[dbo].[shot],[dbo].[scene]
  where  object.frameid=frame.frameid and
frame.shotid=shot.shotid and shot.sceneid=scene.sceneid and
objname=@obj2
select inside1.sceneid as param1,inside2.scpath as param2
from inside1,inside2  where (inside2.maxy -
inside2.miny)/(inside2.maxx - inside2.minx) =
  (inside1.maxy - inside1.miny)/(inside1.maxx - inside1.minx)
and inside1.sceneid=inside2.sceneid
```

Declaration

I, the undersigned, declare that this research is my original work and has not been presented for degree in any other university, and that all sources of materials used for the research have been acknowledged.

Declared by:

Name: Endris Osman

Signature: _____

Date: _____

Confirmed by advisor:

Name: Fekade Getahun (PhD)

Signature: _____

Date: _____

Place and date of submission: Addis Ababa University, January 12, 2017.