Natural Language Engineering **7** (1): 1–27. Printed in the United Kingdom

© 2001 Cambridge University Press

Natural language interface to video database

T. R. GAYATRI and S. RAMAN

Department of Computer Science and Engineering, Indian Institute of Technology, Madras - 600 036, India
e-mail: raman@iitm.ernet.in

(*Received 15 August 1997; revised 13 November 2000*)

Abstract

In this paper, we discuss a natural language interface to a database of structured textual descriptions in the form of annotations of video objects. The interface maps the natural language query input on to the annotation structures. The language processing is done in three phases of expectations and implications from the input word, disambiguation of noun implications and slot-filling of prepositional expectations, and finally, disambiguation of verbal expectations. The system has been tested with different types of user inputs, including ill-formed sentences, and studied for erroneous inputs and for different types of portability issues.

1 Introduction

Access to voluminous video data like that of games, movies, etc., is rapidly increasing due to the information explosion in the repository of the World Wide Web (WWW) (Juliussen 1997). The design and management of a database containing such non-traditional data is a challenging task. Unlike conventional database items, video data cannot be easily represented by a fixed set of common properties or attributes. This is because video data represents real world events, and hence the contents are very diverse in nature (Sridharan and Raman 1996*b*). However, humans are adept in understanding the contents and deriving the attributes, but not necessarily in knowing what they are or how to express them precisely. Hence, a video data model should focus on the representation of video contents to facilitate *content-based retrieval* (Gudivada and Raghavan 1995).

To enable content-based retrieval, comprehensible descriptions of contents of video have to be provided in the form of metadata to video data (Jain and Hampapur 1994). Content-based retrieval mainly focuses on high-level contents (Leung, Hibler and Mwara 1994). Since state-of-the-art image processing techniques are not yet capable of identifying the high-level contents, the video database systems that have been developed so far represent the high-level contents using some form of textual description. Hence, any user interface is essentially mapping the user requirement to the meta-database of textual descriptions. In this paper, we discuss a natural language interface to such a meta-database of structured textual descriptions of video objects.

https://doi.org/10.1017/S1351324901002601 Published online by Cambridge University Press

Natural language interfaces to traditional databases have so far not proved very effective in any practical application (see the Editorial to the first edition of *Natural Language Engineering* journal in 1995). Either the query had to be reformulated for the system to understand correctly and respond, or the domain of application had to be severely constrained for the purpose of understanding. Hence, a natural language query interface was not very attractive and successful.

However, in the case of content-based retrieval from non-traditional databases such as video databases, the end-user may often be a naive user. This is especially the case when the database resides in the information repository of the World Wide Web, of which the most important constituent is the naive user. Hence, a natural language user interface with a knowledge-assisted retrieval system may not only be a useful option, but may also be mandatory in certain applications because of the diversity in the contents of the data.

The natural language interface discussed in this paper is built as a front-end to a database of structured descriptions that are in the form of annotations of video objects. The database is based on an annotation data model proposed by Sridharan and Raman (1996*a*). The natural language input that is mapped to this structured annotation database includes ill-formed text in English. Here, we present a method for natural language analysis based on the principles of conceptual analysis. In the following section, we give a brief review of some work related to application of natural language processing in video and image applications. In section 3, we explain the annotation schema for video data, and in section 4 we discuss the natural language interface design in detail . This section consists of an overview of the conceptual analyser as developed by Birnbaum and Selfridge (1981), followed by a discussion on the simplifications of some aspects of this analyser, the design of lexicon for the proposed analyser, the analysis process, and the results.

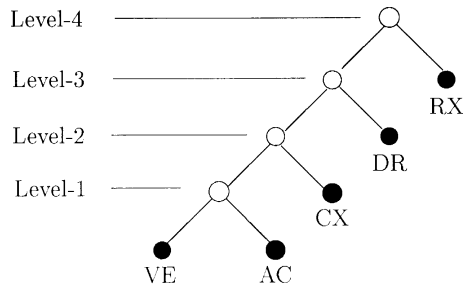
2 The video annotation model

This model defines a generic data structure for representation of high-level contents of video data. Even though there can be many data structures for the same application, there are some fundamental features that the structure should reflect. The data structure for video data that will be described below reflects video data's fundamental characteristics. These characteristics are generic to video data whatever the domain and application.

Here, a video object is defined to be a sequence of meaningful scenes. The video data is viewed as a sequence of real world events which are made up of a set of *elements* and *actions* related to each other under some *context*. There is often something that can be *derived* and *reasoned out* from the actions involving elements in the sequence of events.

The data structure that describes the video data is a collection of concept hierarchies of the five video data categories mentioned above, and their semantic associations organised at multiple levels. A five-level abstraction technique is employed for extracting categories from video data. The data structure is unique, as its meaning and organisation are based on the appealing characteristics of video

https://doi.org/10.1017/S1351324901002601 Published online by Cambridge University Press



VE - Video Element level category set
AC - Action level category set
CX - Context level category set
DR - Derivation level category set
RE - Reasoning level category set

Fig. 1. The Language-based annotation model.

data, namely *richness of information content, rapid context switching, and spontaneous reaction content* (Sridharan and Raman 1996*b*).

The five-level abstraction used to represent the video data is as follows: *Video element* level (VE); *Video element action* level (AC); *Context* level (CX); *Derivation* level (DR); and *Reasoning* level (RX).

A *video element* is a primitive entity around which the entire organisation and retrieval are centred. At the *video element* level abstraction, categories relating to the video elements along with their attributes are extracted. The *video element action* level abstraction considers the actions involving the video elements. Considering the information content in video data in terms of interaction, the representation at these two levels may not be complete. Hence, to enhance the information content and also to provide the best natural interaction, the other three levels of abstraction have been employed.

The binary association among the five categories is shown in the form of a tree in figure 1. It portrays the data structure of the video data. The five leaf nodes contain the taxonomy of categories and category classes organised as generalisation hierarchies, and the four non-leaf nodes represent the interfaces and interactions of the categories. Each of the levels, starting from the lowest level of non-leaf node and moving progressively upward to every non-leaf node, indicates the increasing knowledge level of the data that it represents. Level 1 corresponds only to the information requirements with respect to the Video Elements (VE) and their actions (AC). Level 2 consists of details about ideas, situations, events or information about VE and AC nodes. Level 3 contains conclusions about Level 2 categories, and Level 4 explains how this conclusion process is arrived at. These levels are also indicators of the quality of service that the data structure can provide with respect to the expectation of the end users. Figure 2 illustrates the concept hierarchies corresponding to an application, namely a video database of cricket scenes. The illustrated concept hierarchies are nothing but generalisation (*is_a*) hierarchies.

https://doi.org/10.1017/S1351324901002601 Published online by Cambridge University Press

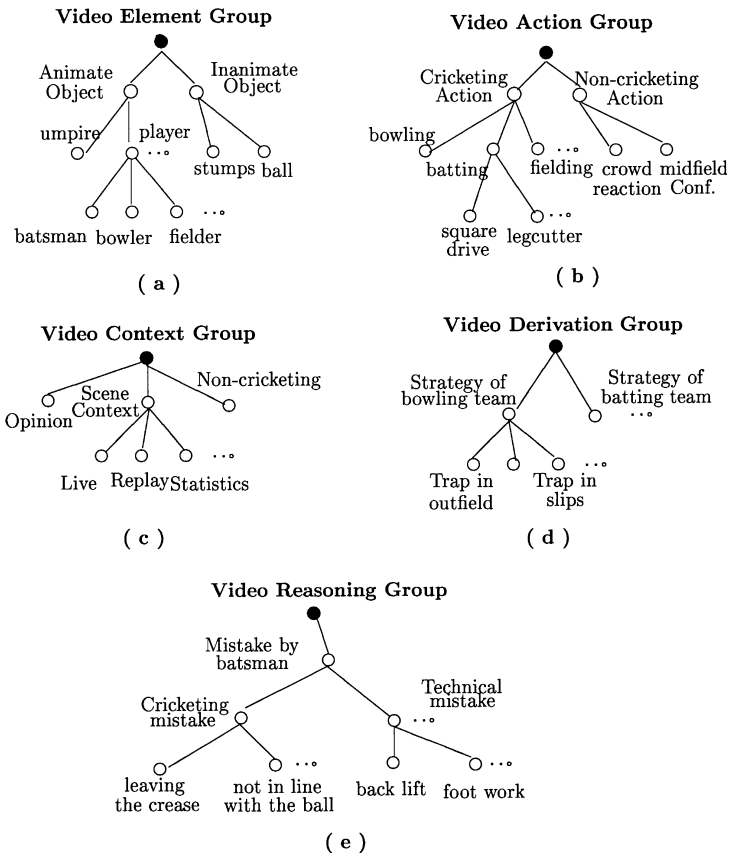


Fig. 2. Illustration of the five concept hierarchies related to the Language-based Annotation Model. (a) Element hierarchy; (b) Action hierarchy; (c) Context hierarchy; (d) Derivation hierarchy; (e) Reasoning hierarchy.

In the video database, an *interval* is represented by a pair of numbers, a starting frame number and ending frame number, and it denotes a continuous sequence of video frames. A *video object* is a sequence of meaningful video frames (scenes) that have their own attributes and attribute values describing the contents. A *video session* consists of many video objects corresponding to a given elapsed time. Many video sessions constitute a *video clipping*.

In the process of formally defining a *video-object* (Oomoto and Tanaka 1993), we assume the following mutually disjoint (countably infinite) sets:

- A set $\mathscr{D}$ of atomic values such as numbers, names and *nil* value,
- a set *ID* of *object identifiers*,
- a set *IN* of *intervals*,
- a set *C_Ve* of *video element* category names,
- a set *C_Ac* of *action* category names,
- a set *C_Cx* of *context* category names,
- a set *C_Dr* of *derivation* category names, and
- a set *C_Re* of *reasoning* category names.

https://doi.org/10.1017/S1351324901002601 Published online by Cambridge University Press

A *video-object* is a triple $(oid, \mathscr{I}, ANN)$, where

1. *oid* is an object identifier $\in ID$,
2. $\mathscr{I}$ is a finite subset of IN,
3. ANN is the language-based annotation given by the ordered set, $ANN = \{(\text{Ve_Ac_Cx_Dr_Re})_i | 1 \leq i \leq n$, and $n =$ number of element-action associations in the video-object $\}$.

Ve, Ac, Cx, Dr, and Re are represented by category-attribute pairs: Ve:(c:a), Ac:(c:a), Cx:(c:a), Dr:(c:a), Re:(c:a), where Ve:c $\in C_Ve$, Ac:c $\in C_Ac$, etc., and a $\in \mathscr{D}$.

Example Suppose a video object contains a scene in which a bowler is bowling, a batsman is batting, and a fielder is fielding. Then, a simplified representation of the annotation is given by

$\{<$ bowler : Russel, bowling : outswinger $>$,
$<$ batsman : Logie, batting : lofting-shot $>$,
$<$ fielder : Greenidge, fielding : catching, derivation : trap in the field $>\}$

That is, the bowler Russel is involved in bowling action (namely, an outswinger), a batsman is involved in batting action (namely, a lofting-shot) and the fielder is involved in fielding action (namely, a catching action). It has been derived from the fielding action of catching that there had been a trap in the field, because of which the catching action could take place.

Thus, a video database consists of several video objects containing structured textual descriptions in the form of *annotations*. The retrieval from the video database is done using the textual descriptions to the video objects, and not by processing the video data directly.

3 The video database with NL interface

To build a complete video database management system, several tasks need to be carried out at the configuration stage. Figure 3 illustrates these tasks and their associations diagrammatically.

First, the five concept hierarchies have to be constructed for any new application. The concept hierarchies consist of concept words in the English language. Then, the annotation of individual video objects is done semi-automatically, with the help of the concept hierarchies. In this way, meaningful structured textual description of the events corresponding to a video object are provided. Several such annotations corresponding to several video objects which belong to several video clippings form a meta-database of annotations.

While the annotations themselves form an index to the video database, meta-indexes are constructed over the annotation database to capture the redundancy in descriptions of the videos' contents. These meta-indexes are index DAGs based on *content inclusion inheritance*. (By *content inclusion inheritance*, we mean that the descriptional data of a video object may be shared by one or more of other video

https://doi.org/10.1017/S1351324901002601 Published online by Cambridge University Press

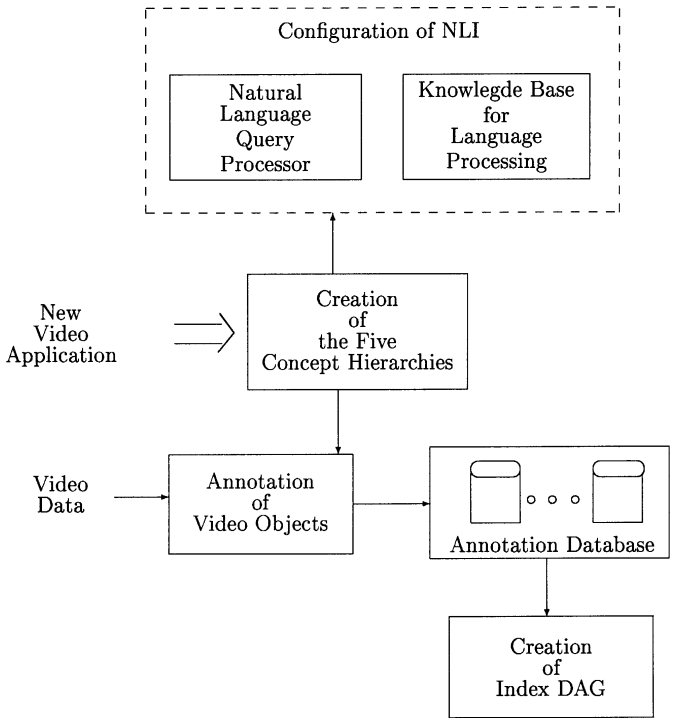


Fig. 3. Tasks involved in creation of an annotated video database system.

objects.) The index DAGs are the result of clustering similar video objects that share the same content.

The principal task of the natural language interface is to extract query information such as *element, action, context, derivation and reasoning* information from the user's requirement specification in natural language. This query information should then be mapped to the appropriate annotation that satisfies the requirement, and the video object corresponding to the annotation has to be retrieved. The natural language analyser that is part of the NLI uses a variant of the classical conceptual analyser for analysing the user requirement input as text in natural language.

Suppose R represents a piece of information requested by the user. Then, the user's requirement may be mapped to one of the following query patterns based on the three boolean query primitives *and, or* and *not*. In other words, the output of the natural language query analyser has to be represented in one of the following standard query formats:

1. Select all/some/any object(s) where $< R >$
2. Select all/some/any object(s) where $< R_1 >$ **and** $< R_2 >$ **and** $< R_3 >$ **and** ....
3. Select all/some/any object(s) where $< R_1 >$ **or** $< R_2 >$ **or** $< R_3 >$ **or** ....
4. Select all/some/any object(s) where $< R_1 >$ **and** $< R_2 >$ **or** $< R_3 >$ ....
5. Select all/some/any object(s) where **not** $< R_1 >$ **and not** $< R_2 >$ ....
6. and so on ....

https://doi.org/10.1017/S1351324901002601 Published online by Cambridge University Press

The input to the database retrieval system is the above-mentioned standard query format. The retrieval from the video database is through the meta-indexing structures.

The overall scheme of the video database management system based on the annotation data model, with the natural language interface, is shown in figure 3.

4 The language understanding system

As already indicated, the analysis of the user input which is a query in natural language, is based on the principles of conceptual analysis. The motivations for choosing conceptual analysis are as described below.

The user's queries with reference to a video database are related to some real world events which are represented by the contents of video. Hence, the natural language query analyser should focus on and understand the event specification in the input query. Of the different methods that have been developed for language understanding, the conceptual analysis mainly focuses on understanding the *event* that is described in a sentence. Conceptual analysis is derived from the theory of Conceptual Dependency (CD), which is concerned with the representation of sentences with respect to the information that they convey about events and states of the world (Schank and Riesbeck 1981). This is one of the main reasons for proposing an analyser based on conceptual analysis.

Another reason for choosing conceptual analysis is the way it considers the role of syntax in language understanding. The major theoretical hypothesis of conceptual analysis is the claim that a separate syntax analysis is unnecessary in language understanding. The arguments behind this hypothesis are as follows:

1. Purely syntax-based approaches have not proved successful in the development of useful understanding systems.
2. It is often sufficient to use a string of a few meaningful words in order for a person to communicate a message to another, which need not be a syntactically complete sentence. With reference to context, the listener can make out the meaning of what is being conveyed. An example of this scenario is the use of telegraphic language.

Evidence against syntax-based approaches can also be seen in Cullingford (1986), which says 'grammar-based' approaches do not easily handle deviant input, including input that is ungrammatical. This is due to the fact that syntax-based approaches tend to be overly concerned with the form of input, rather than with the content.

Thus, the hypothesis that a separate syntax analysis phase is not necessary also makes it convenient for us in accepting the ill-formed text inputs from the end-user.

However, this hypothesis does not imply that syntactic knowledge is unnecessary in the language understanding process. In fact, the conceptual analyser relies on the word order in the natural language input. The very sense of 'word order' implies that some syntactic constraint is satisfied by the words in a sentence. Even in a telegraphic language, we give importance to word order, though we use the minimum number of words sufficient enough to convey the meaning. The semantics may be lost if

https://doi.org/10.1017/S1351324901002601 Published online by Cambridge University Press

word order is not preserved. Hence, a judicial use of syntax information is necessary in an understanding system. We shall refer later to this again, when we discuss the system in detail.

5 Proposed system of analysis

The conceptual analyser (Schank and Riesbeck 1981) is for the general purpose of analysing texts in natural language. In our proposed system, we can go in for some simplification, as there will be no need for an in-depth conceptual analysis, as indicated above.

In our system, the database contains keywords which may be identified by the query processor analysing the natural language input. Since the systems based on keyword search or pattern matching are not complete failures, an input analysis can supplement the keyword identification and need not avoid it totally. Also, in the case of a Natural Language Interface (NLI) to a database system, it is sufficient to extract information with reference to the database requirements. One need not carry out further interpretations leading to deeper understanding. On the other hand, in the case of applications such as machine translation, the analysis must be more rigorous. In these cases, when a natural language L_1 is analysed and translated to another natural language L_2, even the minute details of syntax in L_1 have to be meticulously interpreted and represented in L_2. Thus, there is a little more freedom in the case of analysis for NLI applications.

Another reason for bypassing a rigorous analysis is to simplify the implementation, and thereby meet certain software engineering requirements. Current research in natural language processing is towards the development of practical systems that satisfy the requirements of code reusability, maintainability, system portability (Androutsopoulos, Ritchie and Thanisch 1995) and of fast system development (Cowie and Lehnert 1996).

Thus, in the development of the natural language interface to the database of annotated video objects, we assume the following simplifications:

1. In many applications, representation of semantics at a granularity as fine as that of primitives like INGEST, GRASP, EXPEL, etc., is not needed. Hence, in the proposed system, we do not use the primitives originally proposed by Schank.

   Another point is that the expectations implemented as test-action pairs of requests for a particular word basically examine the occurrence of words representing specific semantic concepts, preceding and following it.

   Hence, in the absence of the original CD structures in our system, we propose the use of the concepts associated with the five concept hierarchies defined by the annotation data model, and devise the expectations for different words based on those concepts. The expectations for a particular word, as encoded in the conceptual analyser, basically are those of certain concepts preceding and following it.

   Though we do not use the CD structures as such, the idea of focusing mainly

https://doi.org/10.1017/S1351324901002601 Published online by Cambridge University Press

on *actions* is held intact, as is done in conceptual dependency. Hence, corresponding to the words that represent the action concepts, we store the Actor and Object expectations. The other information needed for understanding like Direction is captured through the expectations stored for other categories of words which do not represent the action concept.

The generality of representation of the meaning is lost to some extent as expectations are coded in terms of concepts from the concept hierarchies which are application-specific. However, since it is fair to assume that any practically useful system as that of a natural language interface to a database is bound to be application-specific, this loss of generality can be permitted.

Representational convenience plays a major role in our approach. The building of the lexicon for any new application is made much easier than in the original approach. The understandability of the lexicon is straightforward, as it requires the user configuring a new system to know only the different usages of a word.

2. The algorithm of analysis in the case of a conceptual analyser is a combination of top-down and bottom-up processing. That is, the concept structure extraction, noun-group processing, and slot-filling are all done as part of a single procedure or in a single 'pass'. However, we carry out the analysis in three passes. The first pass involves extraction of expectations corresponding to each word in the input text. The second pass involves processing of nouns, noun-verbs, disambiguation of noun implications, and disambiguation and slot-filling of prepositional expectations. The third pass involves slot filling in the Actor and Object slots, corresponding to the *Action* concepts. This simplification drastically reduces the complexity of the code and data structures used in the system. It may seem that the processing time increases due to this method. However, since the query interface does not deal with a large amount of text, the resulting overheads are negligible.

6 System details

6.1 Organisation of lexicon

A syntax parser, in the traditional sense, means a parser based on grammatical expectations in terms of part-of-speech, phrase markers, etc. A specific grammatical pattern is expected to be satisfied by all the words taken together in a sentence. If the sentence is found to satisfy the expected pattern, the syntactic information about the words in the sentence is available as part-of-speech information. However, the grammatical pattern of occurrence of words is difficult to predict in advance in the case of ill-formed text.

The traditional view of syntactic knowledge does not involve word *meaning*. However, the conceptual analyser takes a different view of syntactic knowledge. It is considered as the knowledge of combining word meanings based on their positions in the utterance. While the traditional meaning of positional information implies part-of-speech information, positional information in a conceptual analyser is encoded in the form of word usages strictly based on the meaning of utterance. Hence, for

https://doi.org/10.1017/S1351324901002601 Published online by Cambridge University Press

creating a lexicon, the different word usages in the application domain based on word order have to be analysed first. The lexicon design that has been proposed here considers the different word forms like noun, verb, etc., for a particular word, and encodes the meaning of a word in the form of *expectations* of occurrence of certain concept words along with different word forms of the word, and *implications* in terms of the application-specific concepts.

Though a lexicon tries to capture most of the possible word usages, syntactic expectation rules have to be applied to derive the meaning of certain groups of words like noun groups. Here, syntactic analysis is carried out on small word groups in a local sense, rather than in a global sense which involves syntax analysis of the whole sentence construct.

The following is a detailed description of the configuration of the lexicon.

6.1.1 Representation of verb usage

The verb form of a word represents some action pertaining to some event. Hence, we retain the concept of Action-Object expectation from the CD concept of Event. That is, corresponding to a word that represents a verb, the following items are stored in the lexicon: the concept or the word that it expects in its actor slot; the concept or the word that it expects in its object slot; and the implication in terms of *Action* concept. By the term 'expectation of concepts' we mean expectation of concept words as given by the concept hierarchies of the annotation data model. There can be more than one Actor-Object expectation corresponding to a specific verb.

Figure 4 shows the lexical entries for some of the verb usages. The Actor-Object entries represent the verb usage in the sense of active voice. The entries taken in reverse order will represent the verb usage in passive sense. We specifically mention this because the expectations are dependent on word order.

6.1.2 Representation of preposition usage

Similar to the representation of verb usage, the prepositional usage is represented as *predecessor-successor* pairs of words or concepts, corresponding to prepositions used in natural language text related to the application. Figure 5 shows the lexical entries for some of the preposition usages. In a way, this representation encodes the Direction information that is supposed to occur with Actor-Object expectations.

6.1.3 Representation of nouns

Nouns are represented as words and their corresponding implications in the context of some application. A notable aspect is that the concept words given by the concept hierarchies of the annotation data model may themselves occur in noun form . Thus, noun representation includes these concept hierarchies, as well as some other word forms that are not directly related to these concept hierarchies.

For example, consider the word 'name'; it may reflect a team-name, a player-name or an umpire-name. While player-name is part of 'player' entry in concept hierarchy

https://doi.org/10.1017/S1351324901002601 Published online by Cambridge University Press

verb : stop
Number of Expectations : 2

Expectation	Actor	Object	Implication
1	e_fielder	e_ball	a_stopping (a_fielding)
2	a_rain	x_match	a_abandon-match

verb : miss
Number of Expectations : 2

Expectation	Actor	Object	Implication
1	e_fielder	a_runout	a_runout-miss (misfielding)
2	e_fielder	a_catch	a_catch-drop (misfielding)

verb : loft
Number of Expectations : 1

Expectation	Actor	Object	Implication
1	e_batsman	e_ball	a_lofted-shot (a_batting)

Fig. 4. Lexical entries for some verb usages (the prefixes e, a and x indicate element, action and context categories, respectively).

for *Elements*, and team-name is part of 'team' entry in the concept hierarchy for *Context* information, the word 'name' is not part of any concept hierarchy. Thus noun representation apart from the words in the concept hierarchies, must add word 'name' and its implications such as team-name, player-name, and umpire-name to the lexicon.

6.1.4 Noun-verb representation

Depending on the usage, a word may have either a noun or a verb sense. Putting it differently, certain verbs may occur as nouns in which case some other word in a given sentence plays the role of a verb. A list of such occurrences is maintained for reference.

For example, consider the word 'catch'. It occurs as a verb in sentences such as "The fielder caught the ball", "The batsman was caught by the fielder", etc. However, it occurs as a noun in sentences such as, "He took the catch", "He gave a catch to the fielder", etc. In such usages, we store the verbs that get affected by other verbs. That is, corresponding to the verb 'give', we store affected verb entry as 'catch'. During analysis of the sentence, say, "He gave a catch", the system will first analyse 'catch' in its verb sense. However, it is forbidden from doing so as there is an occurrence of 'give' which has only the verb sense. Thus, 'catch' is categorised as noun and the implications corresponding to the noun sense of "catch" are retrieved.

https://doi.org/10.1017/S1351324901002601 Published online by Cambridge University Press

preposition : at
Number of Expectations : 2

Expectation	Predecessor	Successor
1	e_fielding	x_field-position
2	e_fielder	x_field-position

preposition : by
Number of Expectations : many

Expectation	Predecessor	Successor
1	a_batting	e_batsman
2	a_bowling	e_bowler
3	a_umpire-decision	e_umpire
	. . .	

preposition : off
Number of Expectations : 3

Expectation	Predecessor	Successor
1	e_ball	e_bowler
2	e_ball	e_pad (of batsman)
3		e_bat (of batsman)

Fig. 5. Lexical entries for some preposition usages (the prefixes e, a, and x indicate element, action and context categories, respectively).

6.1.5 Similar/related words information

Different word forms of certain words do not affect the meaning or information that is extracted from the text. Hence, these word forms are replaced by a common word so as to facilitate a consistent representation of expectations involving these words.

Certain words in singular and plural form, certain words in different tense forms, words that have synonymous meaning are all examples of different word forms that do not affect the inference of meaning.

6.2 The analysis process

The prime focus of the process of analysis is to disambiguate among several meanings of a word. The ambiguity in word sense occurs when there is more than one expectation or implication corresponding to a word. The analysis process tries to disambiguate with the help of meaning implied by the neighbouring concept words. The term 'neighbouring word' basically implies the words that precede or succeed the

https://doi.org/10.1017/S1351324901002601 Published online by Cambridge University Press

word of interest. However, under certain circumstances, this condition is violated in which case the words of related semantics are considered for disambiguation purpose. In that case, the words that do not necessarily precede or succeed the word of interest may also have to be considered.

The more carefully we design the lexicon, the easier will it be to process the natural language input. The algorithm consists mainly of three passes, as was mentioned earlier. A detailed description of these three passes is given below.

The First Pass

The steps in the first pass of analysis are as given below:

1. Scan words from left to right.
2. Find the category of the incoming word.
3. If the word belongs to the noun category, retrieve the implications, and add them to the Expectation/Implication List (EIL). Mark the word category as 'noun'.
4. If the word is an *Element*/*Action*/*Context* concept word, add the word to the EIL and mark the category appropriately.
5. If the word belongs to 'verb' category, retrieve 'Actor-Object' expectations and the corresponding implications from the lexicon and add them to the EIL. Mark the category of the word as 'verb'.
6. If the word belongs to the 'preposition' category, try to match the preceding word with one of the 'predecessors' corresponding to the preposition in the lexicon. If there is a match, retrieve the corresponding predecessor-successor expectations and add it to the EIL. If there is no match, retrieve the default set of predecessor-successor expectations and add them to the EIL. Mark the word category as 'preposition'.
7. If the word is a proper noun, find out the category of the proper noun, and add the word to the EIL and mark the category as 'Propernoun'. (In this application, categories of proper noun are player names, umpire names, team names, etc.)

The Second Pass

The module implementing the second pass first identifies the words in their verb form acting as nouns. It marks them as nouns, and their implications as nouns are extracted from the lexicon. Next, the analysis process disambiguates the implications of certain noun words which have more than one meaning. In order to disambiguate, the neighbouring concept words of the ambiguous noun word are examined to find out if their implications disambiguate the meaning of this noun. Similarly the successor expectations of prepositions are disambiguated.

The Third Pass

In the third pass, the task of disambiguating and filling the slots of Actor-Object expectations is undertaken. When there are more than one Actor-Object expectations corresponding to the word of category 'verb'", we assign weights to the expectations, in order to resolve ambiguity. Initially, all the expectations are assigned a weight of zero. 'Actor' refers to the word that precedes the verb in active sense. Thus, if

https://doi.org/10.1017/S1351324901002601 Published online by Cambridge University Press

the expected word or concept is found to have occurred just before the verb, we increment the weight of 'Actor'. Similarly, 'Object' generally means concept word that follows the verb. The succession of an expected concept or word leads to an increment in the weight of the 'Object'. The Actor-Object expectation with the highest weight is chosen to be the valid one.

The representations of linguistic knowledge and concept hierarchy discussed so far are generic to most of the video applications. However, sometimes the application and the type of user input may necessitate some application-specific knowledge to be encoded in the system, apart from the linguistic knowledge and concept hierarchies. This is illustrated below with examples.

Domain Dependent Knowledge

Consider the phrase, "catches given by wicket-keeper of England". This phrase actually refers to the wicket-keeper getting out due to catching, when he bats. However, there is neither the direct mention of the word 'batsman' nor the appropriate player name. The player name has to be inferred indirectly, in which case the structured query will be

Select all objects where
{< element:batsman = Select *player_name* where
{< element:player = wicket-keeper >
< context:team = England >} >,
< action:fielding = catching >}

Thus, in this case, the domain knowledge is also needed to translate the query requirement to the structured form correctly.

Similarly, consider the example "sixers and boundaries of Greenidge". An inappropriate query that will be generated due to this natural language input is

Select all objects where
{< element:batsman = Greenidge >
< action:umpire-decision = sixer >
< action:umpire-decision = boundary >}

This interpretation is not correct, because there cannot be a scene in which both a sixer and a boundary are scored. Hence, one of the correct interpretations is obtained by splitting this query into two queries, namely

Select all objects where
{< element:batsman = Greenidge >
< action:umpire-decision = sixer >}
and
Select all objects where
{< element:batsman = Greenidge >
< action:umpire-decision = boundary >}

https://doi.org/10.1017/S1351324901002601 Published online by Cambridge University Press

Punctuation

Punctuation has been made use of, to some extent. A full-stop or semi-colon is considered to be a break point in the text. Apostrophes are treated as follows: when a noun has an apostrophe such as in a usage say, "Logie's batting", the noun phrase is converted as "batting of Logie" and processed. When an apostrophe is associated with a number, as in the case of the usage "World Cup '96", it is checked for reference to a year or date detail associated with some name of an event such as a Series which is World Cup, in this example. Comma is interpreted as 'and'. Usage of comma in any other sense is not taken care of.

7 Implementation

The proposed approach to the conceptual analysis of natural language text has been implemented on an experimental basis with a sample knowledge-base consisting of a few widely understood concepts in a Cricket application. Though conceptual analysis is powerful enough to process many types of complex sentences, as a pilot study the experimental system was developed and tested to process the following subset of natural language text. The subset consists of simple sentences, including prepositional phrases, in both active and passive voices, sentences with conjunctions such as *and* and *or*, incomplete sentences in the form of phrases, telegraphic language sentences which are meaningful sequences of words with no strict obeyance of syntax, multiple sentences consisting of any of the above mentioned forms, but without complications such as anaphora and ellipses. In all, the total number of sentences with which the system was tested amounted to 250. The analysis of a natural language text using the proposed system is illustrated with the help of examples given below.

7.1 Examples

Simple sentence

"Show some video clips of catches taken by Greenidge".

Here, the user requirement has been input in the form of a complete sentence. The first few words of this query namely "show some video clips of" can be directly mapped to the standard query format

"Select some objects where $< R >$"

where R represents the information requirement pertaining to element/action/context/... information that has to be extracted from the NL text.

Next, we describe the semantic analysis of the sequence of words, "catches taken by Greenidge". Let us consider the first pass of the analysis process. The *expectation/implications* List (EIL) that is obtained after the first pass is given below. (The prefix 'c' is used for the concept words belonging to the concept hierarchies; otherwise, the prefix 'w' is used).

https://doi.org/10.1017/S1351324901002601 Published online by Cambridge University Press

Word: *catches*
Category : *verb*
Expectation : Actor (Pred): c_*fielder*
Object (Succ) : c_*batsman*
Implied Action: c_*catch-out*

Word: *taken*
Category: *verb*
Expectation-1 (in passive sense):
Object (Pred): w_*catch*
Actor (Succ): c_*fielder*
Implied Action: c_*catch-out*
Expectation-2:
Object (Pred): w_*wicket*
Actor (Succ): c_*bowler*
Implied Action: c_*out*

Word: *by*
Category: *preposition*
Expectation: Succ: c_*player* / c_*team*
(Corresponding to predecessor 'taken')

Word: *Greenidge*
Category: *proper noun*
Implication: c_*player_name*

During the second pass, the word 'catches' is understood to be a noun due to the occurrence of the verb 'taken', and hence, its implication corresponding to the verb 'taken' is 'catch-out'. The EIL entry for the word 'catches' is thus modified as follows.

Word: *catches*
Category: *noun*
Expectation: -
Implication: c_*catch-out*

The disambiguation of expectation of preposition 'by' is done next. The disambiguity between the occurrence of the 'player' concept and 'team' concept has to be resolved. Since the successor concept 'player_name' is part of the 'player' concept, 'player' is assigned the proper noun 'Greenidge' which is the name of a player. That is, the resultant of this disambiguation is the assignment 'player = Greenidge'.

After pass-2, the EIL has the following information:

Word: *catches*
Category: *noun*
Form: *plural*
Expectation: -
Implication: c_*catch-out*

https://doi.org/10.1017/S1351324901002601 Published online by Cambridge University Press

Word: *taken*
Category: *verb*
Expectation-1 (in passive sense):
Object (Pred): w_*catch*
Actor (Succ): c_*fielder*
Implied Action: c_*catch-out*
Expectation-2:
Object (Pred): w_*wicket*
Actor (Succ): c_*bowler*
Implied Action: c_*out*

Word: *by*
Category: *preposition*
Implication: c_*player* = *Greenidge*

Word: *Greenidge*
Category: *proper noun*
Implication: c_*player_name*

In pass-3, the Actor-Object expectations corresponding to the verb 'taken' are disambiguated and the slots are filled appropriately. The Action-Object pairs acquire weights depending on the satisfaction of their expectations, as was mentioned earlier. Accordingly, the 'catch-fielder' expectation acquires a weight more than the 'wicket-fielder' expectation. Hence, 'catch-fielder' expectation is inferred to be valid, and accordingly the slot corresponding to 'fielder' expectation is filled with 'fielder = Greenidge' information. After pass-3, the following inference is available:

< element:fielder = Greenidge >
< action:catch-out >

Applying knowledge about the application domain further that a catch-out action associated with the fielder implies a fielding action of catching by the fielder, we arrive at the standard form of the query as:

Select some objects where
{< element:fielder = Greenidge >,
< action:fielding = catching >}

Sentence with conjunctions

Consider the following input of a sentence with the conjunction 'and':

"display catches taken by Greenidge and catches given by Greenidge".

The system takes the conjunction as a break point and comes out with inference for the sequence of words up to that position in the sentence. However, this procedure will fail in the following cases: Example (a): "catches taken and given by Greenidge"

Example (b): "display all boundaries and sixers declared by Bird"

https://doi.org/10.1017/S1351324901002601 Published online by Cambridge University Press

This is because of specification of two actions by a single actor. To overcome this problem, certain rules of syntax and semantics are to be maintained.

Example (a) will be dealt with as follows. When a verb form 'given' is encountered immediately following 'and', it has to seek its Object from the words preceding 'and'. Since the word 'catches' forms the Object for the verb form 'taken' *also*, and since 'taken' does not find its subject in the set of words preceding 'and', it seeks the Actor from words following 'and'. More precisely, the rule corresponding to this usage is *if 'and' is flanked by two participle verb forms, the Object and the Actor are shared by them.* The second example (b) is more tricky. In the first example, syntactic incompleteness in the form of a participle verb form followed by 'and' helped us to frame a general rule based on syntax. However, such a possibility is not there in the second example. Hence, we take the aid of semantics to solve the problem.

The query by itself lends two ambiguous interpretations. However, the most common one is "display all boundaries declared by Bird", and "display all sixers declared by Bird". To derive this interpretation we adopt the following reasoning: if the conjunction 'and' is flanked by the same concepts, of the form C1 and C2, and if C2 has attributes while C1 does not have any, then, C1 inherits the attributes of C2. That is, initially the output of the understander module is:

Select all objects where
{< action:umpire-decision = boundary >
and
< action:umpire-decision = sixer >
< element:umpire = Bird >}

Since < *umpire-decision* = *sixer* > has the attribute information < *umpire* = *Bird* >, < *umpire-decision* = *boundary* > also shares it, and hence the above output is modified into a set of two queries,

Select all objects where
{< action:umpire-decision = boundary >
< element:umpire = Bird >}
and
Select all objects where
{< action:umpire-decision = sixer >
< element:umpire = Bird >}

A similar example is *display all boundaries and sixers by West-Indies*, in which case the NLU output is

< umpire-decision = boundary >
and
< umpire-decision = sixer and team = West-Indies >

This has to be modified as explained previously.

https://doi.org/10.1017/S1351324901002601 Published online by Cambridge University Press

Phrase

The processing proceeds on the same lines as those for simple sentences, except that the query format "show me all the clips that satisfy the requested information" is assumed.

Telegraphic language

Consider the following input:

"Greenidge bowling catching mid-on"

Let us do a brief analysis of the above input. The *expectation/implications* List (EIL) that is obtained after the first pass is as given below:

Word: *Greenidge* Category: *proper noun*
Implication: e_*player_name*

Word: *bowling* Category: *verb*
Expectation-1 (in active sense):
Actor (Pred): c_*bowler*
Object-1 (Succ): e_*batsman*
Object-2 (Succ): a_*type of bowling*
Implied Action: a_*bowling*

Word: *catching*
Category: *verb*
Expectation-1 (in active sense):
Actor (Pred): c_*fielder*
Object (Succ): e_*batsman*
Implied Action: a_*catching*

Word: *mid-on*
Category: *context*
Implication: x_*fieldposition*

There is no ambiguity about nouns and noun-verbs in this example. Hence, there is no change in the list during pass-2. In pass-3, there is no disambiguation process, as there are no ambiguous expectations. Only slot filling is to be done. The 'bowler' expectation of the verb 'bowling' is satisfied by the player name 'Greenidge'. Corresponding to the verb 'catching', there is no information available about the 'batsman' or 'fielder', except the implication of the *action* 'catching'. The word 'mid-on' is inferred to be 'field-position' information. Therefore, the output is

Select all objects where
{< element:bowler = Greenidge >,
< action:fielding = catching >
< context:field-position = mid-on >}

https://doi.org/10.1017/S1351324901002601 Published online by Cambridge University Press

Multiple sentences

Applying the same technique of analysis as described for the above-mentioned cases, queries involving multiple sentences are also analysed, taking the punctuation marks such as full-stop, semi-colon, and question mark as break points for analysis. The extracted information is expressed in a structured query format. For example, consider the input

"There was a match between India and Pakistan in World Cup '96. Pakistan's Amir Sohail was stumped out in that match. Is there a picture of that stump-out (scene)?"

After extracting information from the individual sentences, the structured query output will be

Select any object where
{< element:batsman = Amir Sohail >,
< action:fielding = stump-out >
< context:match = India/Pakistan >
< context:series = World Cup >
< context:year = 1996 >}

The above examples illustrate the effectiveness of the proposed modified Conceptual Analysis in dealing with different types of input sentences. It can also be seen that successful analysis could be done by using higher level concepts provided by the annotation model, instead of the actual CD primitives.

8 Evaluation of the approach

8.1 Handling of erroneous inputs

For an input consisting of improper sequence of words, the output of the system may also be equally meaningless. It can, however, be overcome to some extent by imposing some semantic constraints on the structured query output from the query processor, and thus ensure that incorrect queries are rejected by the system. However, the interpretation of the input could be different from the original intention.

8.2 Portability issues

8.2.1 Knowledge domain portability

A natural language interface to a database is said to provide knowledge domain portability, if it can be configured for use in a wide variety of knowledge domains without significant changes in the code.

Since the approaches to video data modelling as well as natural language analysis are basically centred around real-world *events* which are caused by *actions* of another set of real-world entities, knowledge domain portability is realised for the class of all action-oriented applications such as games, movies, etc. For other types of applications, such as conference proceedings, or analysis of satellite pictures,

https://doi.org/10.1017/S1351324901002601 Published online by Cambridge University Press

domain knowledge configuration necessitates significant changes in the code. We do not expect our video annotation data model to support those types of applications.

Though not directly related to video databases, it is worth recording our experience with a similar approach to natural language analysis in our efforts to develop a natural language interface to library information system (Gayatri and Raman 1996). The approach could not be easily adapted for an inventory application. It required significant changes in the code of the program on the change of application.

Configuring a natural language interface for a new knowledge domain involves creating or modifying the lexicon, the world model, and the mapping of the concepts in the knowledge domain to the database. Different systems assume that the knowledge domain configuration will be carried out by people having different skills, namely the programmer, or the knowledge engineer, or the database administrator, or the end-user. In the proposed natural language interface to video database, we assume that the programmer will be the appropriate person to configure the system. This is due to the fact that a small part of the code may have to be changed due to application-specific mapping, as was illustrated before. Hence *a priori* knowledge of the code of the natural language interface system is assumed on the part of the user, if he wants to use the system for different applications.

The main part of the world model for a specific application comes along with the five categories of concepts, namely elements, actions, context, derivation and reasoning information, defined in the video annotation model. This is a major facility available to the programmer for configuring the system. Hence, typically, the task of the programmer is to collect different *usages* of words and concepts pertaining to a specific application, and then develop the lexicon based on those usages.

Usually a person who configures a natural language system should be familiar with knowledge representation techniques, linguistic concepts and database representation. In the proposed system, since the representation of the lexicon is quite straightforward without any complexities, the programmer need not worry about complex knowledge representation techniques. However, he should be familiar with the video annotation data model and the linguistic concepts.

8.2.2 DBMS portability

A natural language interface is said to provide DBMS portability if it can be easily modified for use with different underlying database management systems.

The proposed system is portable to different DBMS systems. This is due to the feature that the initial output obtained after conceptual analysis provides an intermediate representation consisting of concepts and their associations related to the application. Though the concepts that have been used in the illustrations pertain to the video data model, they are common to any other real world application as well. Hence, mapping this representation to a database query language can be done differently for different database management systems. In fact, as was mentioned earlier, a variant of the proposed approach to conceptual analysis was used to build a natural language interface to a relational database of library information.

https://doi.org/10.1017/S1351324901002601 Published online by Cambridge University Press

8.2.3 Natural language portability

A natural language interface is said to possess natural language portability if it can be modified for analysing inputs in different languages.

The proposed approach depends mainly upon a fairly consistent word order in English language and not on the satisfaction of strict syntax rules. That is, the *expectations* for each word are basically composed of rules that check for occurrence of concepts and concept words before and after the word. Therefore, it may be possible to use the same approach to other languages which do not deviate much in word order. In that case, the expectations can be stored in the form of predecessor-successor information, as was done in the case of English. However, if the word order is not consistent, then it is not possible to follow the proposed approach as expectations cannot be created reliably.

8.2.4 Hardware and programming language portability

Most of the current systems are developed using the 'AI' languages like Prolog and Lisp. Though they are commonly available now, they are used mainly by the research communities. However, since a practical natural language interface is to be maintained by some person who may not necessarily be a researcher, it is better to develop the system using a language that is popular among programmers in general. Also, the systems developed using Prolog or Lisp are not generally convenient enough to be interfaced with multimedia databases and video repositories, which generally use languages like Java or C++ these days.

Owing to the above-mentioned issues, the experimental system for testing the proposed approach to language analysis was developed using C language, with the hope of developing further and porting the system across different computing platforms.

To summarise, in this approach the general query processing is portable. However, we cannot resolve certain domain-specific issues in general. The knowledge information and the mapping of the knowledge to structured query is not portable. We do not consider this as a limitation because only a generic NLP system requires a generic knowledge information base. A practical NLP system will always pertain to specific knowledge domains. Hence, this approach is suggested with the strong assumption that it will be used in a practical scenario, rather than analysing its suitability in a generic theoretical sense.

8.3 Related issues

8.3.1 Task of making detailed annotations

It may appear that the task of video annotation of a huge repository is a gigantic one. However, for illustration, we would like to refer to the web site *www.cricket.org* which gives live details of some cricket match going on. Here, apart from the display of a score board, a ball-by-ball update of the game is also given. This update is being done by some cricket enthusiasts. Hence, when such a meticulous update on the

https://doi.org/10.1017/S1351324901002601 Published online by Cambridge University Press

game is already available, we consider the task of forming annotations for a large collection of video clips as quite feasible. In fact, from the viewpoint of NLP, the ball-by-ball updates that are available can be processed and annotations generated for specific video clips. A detailed discussion of this is beyond the scope of this paper.

8.3.2 Automatic lexicon acquisition

Based on different possible queries and sentences, a lexicon can be generated automatically as in the case of annotations. However, it needs manual fine-tuning according to application, which is more pronounced in the case of lexicons than in the case of annotations.

8.3.3 Future applications

As mentioned earlier, it is expected that a large collection of video objects will be available on the web with some form of description, as in the case of www.cricket.org. This NL approach can be used as part of a larger project/sofware that involves building up of a search engine that understands NL to some extent. Precisely, this approach can be a component of some search engines for large database systems.

8.4 Improvement over keyword-based natural language analysis

A keyword-based analysis system has been developed by Sridharan (1997) for analysing different types of input queries in natural language. The work reported in this thesis has enhanced the query processor capabilities further so that it overcomes some of the inadequacies in the previous approach.

The query input has been classified into different categories based on criteria such as the information requirement of different types of users, and the complexity of the natural language text used in the query. Since our main concern is with regard to the complexity of natural language text used in the query, we present here the details of the query performance of the previous system with respect to language complexity.

Of the sample set consisting of 250 queries, the information requirement in 193 queries is regarding element, action and context details. The remaining are concerned with derivation and reasoning. We have not considered processing of queries regarding derivation and reasoning because the semantic processing required at those levels warrants a very detailed piece of research work by itself.

Table 1 shows the statistics on the number of processed queries by the keyword-based system. Table 2 compares the results of the proposed modified Conceptual Analyser against the other.

It can be seen from Table 1 that the keyword-based system is weak in retrieval in the case of queries that need semantic processing. The proposed system successfully processes many of the queries that failed in the keyword-based system, in the sample

https://doi.org/10.1017/S1351324901002601 Published online by Cambridge University Press

Table 1. *Performance of the keyword-based retrieval system for a sample set of queries (Class A – Queries that need only simple analysis; Class B – Queries that need semantic processing)*

	Class A	Class B
No. of Queries	74	119
No. of Successful Retrievals	69	98
No. of Unsuccessful Retrievals	5	21

Table 2. *Comparison of performance of the keyword-based retrieval system and the system based on modified Conceptual Analysis (Class A – Queries that need only simple analysis; Class B – Queries that need semantic processing; Succ – No. of successful retrievals; Unsucc – No. of unsuccessful retrievals)*

Implementation prototype	Class A No. of Queries: 74		Class B No. of Queries: 119	
	Succ	Unsucc	Succ	Unsucc
Keyword-based	69	5	98	21
Modified Conceptual Analysis	74	0	111	8

set of 193 queries. We explain below using a sample set of examples the processing of queries that failed in the previous case.

(1) *display all boundaries and sixers*

The keyword-based system failed in this because it tries to retrieve video-objects that contain both boundaries and sixers. Since the video database contains only video of meaningful sequences of events, it is not appropriate to search for an object that contains both a sixer and a boundary. Therefore, it is necessary to check if an information requirement contains ambiguity; in this example, we check whether the requirement contains two or more umpiring decisions and split the query into two as was illustrated earlier.

This type of processing is highly domain dependent, and hence a method for encoding this type of domain dependent knowledge needs to be devised. Since the research prototype developed is a small one, we maintained the above-mentioned information in the form of small scripts describing sequences of meaningful scenes in the video. For example, a sequence of bowling, batting, fielding and umpiring-decision forms a meaningful sequence of actions and none of the actions can occur twice in a sequence of scenes which contain these four actions. However, for larger systems, more detailed encoding method is necessary.

(2) *display all boundaries and sixers declared by Bird*

A detailed explanation for this example was seen in section 6.2.

https://doi.org/10.1017/S1351324901002601 Published online by Cambridge University Press

(3) *wickets of Hooper*

Due to the absence of semantic analysis, the keyword-based system does not know whether Hooper is a bowler or batsman or fielder. Hence, the system retrieves all objects that contain *umpire-decision* = *out*, and *player* = *Hooper* information. This problem is overcome because our system understands that Hooper is a bowler by analysis of the word 'Hooper' with respect to the preposition 'of' and its predecessor 'wickets'.

(4) *how many wides have been bowled in the match?*

The keyword-based system fails here because it cannot understand phrases such as 'how many', which is understood by the proposed semantic analyser, and refers to the final score-board displayed apart from retrieving objects that display the wides bowled.

(5) *show all catches taken by wicket-keeper of West-Indies*

Here, again, the keyword-based system does not know whether the wicket-keeper is referred to as a fielder or a batsman, and hence it fails to retrieve the correct set of video-objects. The proposed semantic analyser correctly understands wicket-keeper to be the fielder and hence retrieval is correct.

Certain queries do fail in the prototype developed. These queries fail because they do not provide information that can be directly mapped to the content of a video annotation. The requirement specification is indirect. Processing of such indirect queries demands storage of certain application specific rules that help in mapping the indirect requirement specification to the annotation. The prototype developed does not provide for storage of such rules. The following examples illustrate some queries that failed.

(1) *what is the score of Gooch?*

It can be inferred that the query demands the display of score-board. However, the process of understanding should go a step further to interpret that the query demands the display of score-board depicting the innings of England as Gooch belongs to England team.

(2) *display scenes containing the captain of West Indies*

In this case, the system should understand that an annotation may not always indicate whether a player in a scene is a captain. Hence, it should retrieve the name of the captain of West_Indies and match it with names of players in annotations to check whether a scene contains the captain.

Though some of the inadequacies in the keyword-based approach have been overcome by incorporation of semantic analysis through a modified Conceptual Analyser, the prototype developed is weak in storing all types application specific rules, as illustrated by the examples.

https://doi.org/10.1017/S1351324901002601 Published online by Cambridge University Press

9 Conclusion

The advent of the World Wide Web has renewed interest in natural language interfaces as the naive user community is very large in size. Another reason is that the retrieval from huge multimedia repositories including video databases is content-based. Hence, there is a need for research on application of classical methods of understanding user queries in natural language, to develop practical systems.

In this perspective, we have proposed a modified Conceptual Analysis for building a natural language query interface to a video database. The video contents are represented using a language-based annotation model. Hence, the system of understanding has been tuned to the annotation model.

It was observed that the classical method of Conceptual Analysis could be modified easily to suit the language-based annotation model for performing semantic analysis of user's query specification in natural language. This could be done by modifying the lexicon in terms of concepts provided by the annotation model, and by simplifying the single pass analysis process into three passes.

A prototype of natural language query processor based on the proposed modified Conceptual Analysis was developed. Its performance was found to be better than the keyword-based system in the processing of queries that needed semantic analysis. Though the prototype accommodates certain application specific rules, it failed to process some queries due to lack of a few other types of rules. This shows that a typical natural language interface should also provide for dealing with various types of application specific rules for dealing with a wide range of queries.

References

Androutsopoulos, I., Ritchie, G. D. and Thanisch, P. (1995) Natural language interfaces to databases – an introduction. *Natural Lang. Eng.*, **1**(1): 29–81.

Birnbaum, L. and Selfridge, M. (1981) Conceptual analysis of natural language. In: R. C. Schank and C. K. Riesbeck (eds.), *Inside Computer Understanding*, pp. 318–353. New Jersey: Lawrence Erlbaum.

Cowie, J. and Lehnert, W. (1996) Information extraction. *Comm. ACM*, **39**(1): 80–91.

Cullingford, R. E. (1986) *Natural Language Processing: A Knowledge-engineering Approach.* Totowa NJ: Rowman and Littlefield.

Gayatri, T. R. and Raman, S. (1996) A hybrid model for nlp-based query processing for library application. *Proceedings Indian Congress on Knowledge and Language, Central Institute of Indian Languages*, Mysore, **1**, pp. 53–65.

Gudivada, V. and Raghavan, V. (1995) Content based image retrieval systems. *IEEE Comput.*, **28**(9): 18–22.

Jain, R. and Hampapur, A. (1994) Representation of video databases. *ACM SIGMOD RECORD*, **23**(4): 27–33.

Juliussen, E. (1997) Computers. *IEEE Spectrum*, Technology 1997: Analysis and Forecast Issue, January: 49–54.

Leung, C. H. C., Hibler, J. and Mwara, N. (1994) Content based retrieval in multimedia databases. *Computer Graphics*, **28**(1): 24–28.

Oomoto, E. and Tanaka, K. (1993) Ovid: Design and implementation of a video-object database system. *IEEE Trans. Knowl. & Data Eng.*, **5**(4): 629–643.

Schank, R. C. and Riesbeck, C. K. (1981) The theory behind the programs: Conceptual Dependency. In: R. C. Schank and C. K. Riesbeck (eds.), *Inside Computer Understanding,* pp. 10–26. Lawrence Erlbaum.

https://doi.org/10.1017/S1351324901002601 Published online by Cambridge University Press

Sridharan, P. N. and Raman, S. (1996*a*) Synthesized data structure for video data: A formal specification. *Proceedings IEEE Int. Conference on Image Processing*, pp. 16–19. Laussane, Switzerland.

Sridharan, P. N. and Raman, S. (1996*b*) Characteristics of video data for signal analysis. *Proceedings Int. Conference on Signal Processing*, Beijing, China.

Sridharan, P. N. (1997) *An Intrinsic Video Data Model for Content-based Information Retrieval*. PhD thesis, Indian Institute of Technology.

https://doi.org/10.1017/S1351324901002601 Published online by Cambridge University Press