CRICKET ONTOLOGY

Instructor :- Prof: Navjyothi Singh

Project Team :-

PV Sai Krishna	***(200402036)***
Abhilash I	***(200501004)***
Phani Chaitanya	***(200501076)***
Kranthi Reddy	***(200502008)***
Vidyadhar Rao	***(200601100)***

Project Description :-

We present a system for semantically searching inside the video recordings of the game, Cricket. For this, we propose **ontology of cricket**.

Video data is growing rapidly on the net, yet there are no systems which can search inside them and index that matter. For **semantic reasoning** capability one has to resort to ontology of cricket domain. We attempted to make a hyper-video standard to allow navigation within the video by annotating the video in terms of the standard terms that give us a visualization of the game, Cricket. To enable semantic searching of those cricket videos, we decompose the entire match (video) into actual **episodes**. This episode describe the atomic level of an event that has happened at some level of granularity. The episodes are described based on the ontological terms developed as mentioned earlier. These give a gist of the happenings in the video at regular intervals. The whole video is well described in those terms (all terms in Appendix). This ontology specific marking of episodes in a cricket video is useful for content based video indexing.

An episode is defined on a linear time and has a start point and an end point. As mentioned earlier, an atomic episode is an event. A whole cricket match (video) can be described as follows: The match starts with the event of Toss between a couple of teams among which one team gets an opportunity to bat first whereas the other to field. Then a sequence of overs are bowled. The episodes of a ball can start with the decision of an umpire signaling the bowler to start and once the bowler delivers it, batsman playing it selecting different shots and runs will be scored and the like until the next ball starts. Even an episode can just comprise of a batsman playing the shot itself and as mentioned earlier, the granularity of each episode depends upon at the level at which it is annotated.

The shots played by a batman can be of different types , on-drive, square-drive, pull, etc... An event can be also an umpire decision which includes No-Ball, Six, Four, Bye etc... All the list of terms are shown in Appendix [1].

The building blocks of developing such a tool are

1) Identify all the classes required to describe the domain of cricket.
2) Identify the class hierarchy among the classes.
3) Provide an interface to tag the videos.
4) Manually tagging a few videos.
5) Indexing the tagged videos.
6) Searching the videos given a query.
7) Displaying only the required frames.

Interface:

To tag cricket videos according to the ontological terms of the cricket a basic user friendly interface is needed.
The interface [2] is made user friendly with Video Display controller and a easy way to navigate between the attributes, their values to tag the episodes with their ontological terms. The Video display controller has play,start ,pause buttons and slider to navigate any portion of the video and tag their instances with the ontological terms. Presently these ontological terms were a simple hierarchy in XML containing the instances of the attributes or the Value references of an event or episode. The interface is made user friendly to tag using the Data Grid menu and Tree Display menu. With this basic interface we were able embed the attribute hierarchy into the episodes.

How to use the interface to tag the cricket videos :-

1. Interface has a video display controller with play pause and stop buttons and a slider used to navigate across the episodes of the video.

2. To annotate the start time and the end times of an episode just click on their values while playing the video.

3. Zoom In and Zoom out are provided to annotate the sub-episodes if any present in the current episode.

e 4. Umpire decision is to be modified only at the "umpireDetails" and not at the "sceneDetails" . Changes will be automated when updated at the umpireDetails.

5. In editing other details ,first click on the value of the action to be edited and select the options from the tree displayed to the right at the appropriate frames.

6. Click the button "show" on the bottom to view the complete details of the episode tagged in XML format.

Annotation Tool :-

Annotation is done in XML format. The relation-ship of inclusion is specified as a parent-child node. The attributes start and end are used to specify the start and end point of the episode in the video. There are different annotation types and attributes also provided as explained earlier based on which the video (in terms of episodes) is annotated and stored in XML format.

Now that we have the interface for tagging we can manually tag the videos. The reason for choosing manual tagging is that the tagged data will be correct. If we choose some automated way of tagging like build a training data set of frames representing one class and then tag the new data using this model, we can some times tag the videos wrongly.

Example: -

If we have frames in the training data representing a ball that goes very high in the air as a "SIX", then if a test data frame comes in which the ball is soaring very high it would be tagged as a "SIX" depending on our training data. But it might as well be a catch.So in order to avoid such vulnerabilities in the system we decided to follow manual tagging approach.

Now using the interface one can tag any video easily. The classes provided will cover 80% of the events that occur on a cricket field. We load a video file into the tagging interface browser and play the video. We have been provided with a "stop,pause,forward and back" options, so that we can jump around in the video whenever required. We play the video and try to capture an event . So an entire event can said as a union of "bowler,batsmen,fielder and umpire the decision maker."

EVENT :-

It is described something that happens at a given place and time. For the cricket domain we describe an event as the union of all the actions that take place from the time when a bowler runs into to bowl a ball to the next ball. The reason for defining it in such a way is that, we felt that every user would like to watch a video completely i.e. even though if he wants to watch sachin hit mcgrath for a SIX, the user would love watching

the entire event i.e. mcgrath running in and then sachin smashing him , rather than just the frames where the ball has pitched on the pitch and sachin hits it. In defining an event in this way the excitement of watching the video is never lost.

So now we try to capture all the possible sub events in an event.

1. First we will in the details of the particular match like "match location i.e. where the match is being held, whether is a day/night,20-20 encounter,stadium name etc." These details need to filled only once for one particular match.

2. Then we need to fill the official details,"Team names, umpires" etc. These details also one needs to fill in only once.

After the first 2 steps are done, we need to tag the video now based on the sub events of an event. For tagging an event we first need to tag what is the start time and the end time of the event, the bowler, the batsmen and the fielder involved in the event.

Now having tagged the players involved in the event we try and define the task accomplished by each player in that particular event.

3. For the Batsmen we need to tag what his preaction stances are like gardening etc. Then we need to tag what is the shot played by him, action of playing the shot and finally the result of the action.

4.Similarly for a bowler also we tag his premarkingactions and then the ball delivered and the result of the ball.

In the case of the batsmen the result of the action will generally involve the actions related to a batsmen like "six,four etc" , where as for a bowler the result of the ball will be "wide, noball" etc.

5.Then we tag the details related to a fielder if involved. Finally the umpire being the decision maker gives the decision of the event that has occurred and the user tags it.

We have manually tagged 2 videos each of length 25 min.

Index and Search :

Indexing :-

The obtained XML files corresponding to the annotated videos are traversed using XML DOM parser. Different classes(class names) associated with the events and their values are extracted. These class names and values extracted are indexed using text based indexer & retrieval system "Lucene".

Lucene is a free/open source information retrieval library, initially developed in Java. At the core of Lucene's logical architecture is the idea of a "document" containing "fields" of text. "Lucene Document(Document)" is the most important data type in the architecture of Lucene. At the abstract level all the operations are done on documents i.e lucene generates an index by adding "lucene documents" to the index and also searches and retrieves the "lucene documents" from the index.

Lucene document is in turn build using some set of fields. The advantage of having fields is, it enables us to restrict our search to a particular section of the document rather than searching the complete document.

Part of XML data generated looks like

```
<node type="sceneDetails">
   <nod action="Start Time" value="00:00:00"/>
   <nod action="Finish Time" value="00:00:25"/>
   <nod action="Id" value=""/>
   <nod action="Title" value=""/>
   <nod action="Batsmen" value="Gautam Gambhir"/>
   <nod action="Bowler" value="Brett Lee"/>
   <nod action="Fielders" value="Adam Gilchrist"/>
   <nod action="Umpire Decision" value=""/>
 </node>
 <node type="batsmenDetails">
   <nod action="PreStanceAction" value=""/>
   <nod action="ShotPlayed" value="CoverDrive"/>
   <nod action="ActionAfterShotPlayed" value=""/>
   <nod action="ResultOfAction" value="Beaten"/>
 </node>
```

The xml parser we wrote extractes class names and values like StartTime-->"00:00:00", FinishTime-->"00:00:25", Batsmen-->"Gautam Gambhir", Bowler-->"Brett Lee", etc

The information associated with each class is added to document as a seperate field to support field specific search.

```
doc.add(new Field("StartTime","00:00:00",STORE.YES,INDEX.TOKENIZED);
doc.add(new Field("EndTime","00:00:25",STORE.YES,INDEX.TOKENIZED);
doc.add(new Field("Batsmen","Gautam
                                Gambhir",STORE.YES,INDEX.TOKENIZED);
etc,
```

Each document in index generated has a total of 20 fields using which the complete information about the event can be inferred.

Searching :-

By default the system we designed, searches for the user given query in the complete document (i.e, all 20 fields). For example if the query given is "sachin" the system retrieves the all events in which sachin is present either as a batsmen, bowler or a fielder.

But one can also search for the events in which the role sachin is a batsmen and this can be achieved by the query "batsmen:Sachin".

Using all the 20 fields generated during the index, one can query the system to get the most relevant events.

Ex 1:-

User need is to retrieve events in which, Sachin hits Flintoff through covers when Kevin slides and its a four.

Query : "+batsmen:Sachin +bowler:Flintoff +fielder:"Kevin Pietersen" +fielderaction:slides +umpiredecision:four"

'+' in the above query specifies that, event "MUST" satisfy the following condition

Ex 2:-

User need : Same as in Ex 1, but batsmen can be either Sachin or Dravid.

Query : "+batsmen:(Sachin OR Dravid) +bowler:Flintoff +fielder:"Kevin Pietersen" +fielderaction:slides +umpiredecision:four"

Ex 3:-

User need : Same as in Ex 2, but the bowler is someone else other than Flintoff

Query : "+batsmen:(Sachin OR Dravid) -bowler:Flintoff +fielder:"Kevin Pietersen" +fielderaction:slides +umpiredecision:four"

The output of this search process is

1. Start time of the event
2. End time of the event
3. Video file in which this event has occurred

The above output can be another module which extracts the video frame and presents it to the user.

'-' in the above query specifies that, event "MUST NOT" satisfy the following condition

Using this ontology we develop a standard for the markup of cricket video. Based on this mark-up standard annotation and indexing of the video is done. Further, the search algorithms used indexes the videos which can be used later for querying and retrieving relevant

FUTURE WORK : -

1) We need to write the ontology representation of the cricket. This will help us in developing a system with a semantic reasoner .

2) To reduce the time spent in tagging the videos,the Interface needs some modifications to add the Reasoner with its own ontology to analyze the logical structure of the tagged sequence of episodes. And also add the episode hierarchy with its own ontology which gives a complete ontology of the cricket to the videos.

3) In the search results we are displaying just the start time and the end time. We need to give the video frames as output.

Appendix:

1) List of terms (hierarchical) (our ~1000 terms prepared by hand)
http://web.iiit.ac.in/~vidyadhar/cricket/updatedcricket/srcview/

All the terms are present in the “src ” folder.

2) Reference to the Interface :
http://web.iiit.ac.in/~vidyadhar/cricket/updatedcricket/cricketontology.html