Artificial Intelligence
Mini project-2
Phase-2

Part 1 and Part 2

This phase consists of 4 parts. The focus of this phase is to build a question-answering system which will consist of 4 modules (each corresponding to one of the 4 parts). For each of the 4 parts, few questions (queries) will be given. But apart from these, there will be questions for each part that will have to be answered during evaluation. So hard-coding should be avoided and your code should be as general as possible.

Dataset

For this phase, the commentary data of all the five ODI matches (for both innings) has to be included. The "phase 1" dataset also has to be used.

Part 1 - Text processing

This part deals with questions to be answered that will require string parsing and text processing of the commentary data.

The commentary information consists of :

1. ball by ball details (so naturally over by over information is captured)
2. for each ball, the information is of the form:
PLAYER1 to PLAYER2, RUNS SCORED / "OUT", DESCRIPTION (of the ball and the way ball was played, etc).
3. comments by different users, during the course of play

So questions based on the following grammar need to be answered.

Rough grammar of questions for part 1:

<full question> : <info about match> <description> <question>
<info about match> : first | second | third | fourth | fifth
<description> : this will contain information about few / all of the following:

1. player hitting Ones / Twos / Fours / Sixes (optional number or "maximum" - may be specified like player1 hit 3 sixes)
2. about the specific over(or there may not be any specific over given)
3. player getting out

4. player bowling wide / no ball (optional number or “maximum” - may be specified, see Q5)

<question> : which ball | which over | who dismissed (although this question can be answered even using scorecard info) | who hit | which bowler

Questions:

1. In the first match, Kohli hit sixes in overs of which bowler(s)?
(here number of sixes and specific over is not specified, so you should retrieve all the instances)

2. In the first match, Williamson was out in which over?
(you will need to search for “OUT” word and find the appropriate over/ball information. In this query the answer would be “NONE” as Williamson was not out, a clever solution would be to first search the scorecard information and if required subsequently search the commentary)

3. In third match, Taylor hit fours in which overs?

4. In fourth match, Dhoni hit six in over 30, in which ball?

5. In third match, Anderson bowled maximum wides in which over?

NOTE: These are sample questions, so you should follow the grammar rather than hard-coding specific instances. Once you have the functions according to the grammar, you have to call them for these examples, and for other questions which will be released at the time of evaluation.

Part 2 - Automatic query generation for “phase 1”

This part is an extension of the first phase (mostly set A) of this mini project (so commentary data need not be used). In the first phase, most of you had created different models (nltk.Model) for different queries by first constructing the string of appropriate predicates. This was followed by manually creating the string of the given query, and then invoking “nltk.evaluate” function to get the corresponding outputs.

For this part, the “query string creation” part need to be automated. As a first step, you need to have a single model of all the predicates that you used for “phase 1”. For this, you may have to modify the predicates that you used (so that a general model can be build). Once you have the complete “model” of all the predicates (like “strikegt” : strike rate greater than 200, etc), given queries need to be automatically converted to appropriate nltk format string. Then “evaluate” function need to be called to get the outputs.

The main task is to read a query and generate the appropriate format string for evaluation by nltk. For conversion, a grammar is needed.

Rough grammar for part 2:

<main question> : <quantifier info> <description 1> <connector> <description 2>
<quantifier info> : “for all matches” | “for all innings” | “there exists player” | “there exists a match”
<description1>, <description2> : will be related to any of the predicates like strikelt : strike rate less than 100, etc
<connector> : “and” | “if - then” | “is given to” | “consists of” | “contains”

Questions:

1. For all matches, player of match award is given to player of winning team.
(so “player of match award” , “player of winning team” will be 2 predicates, and “is given to” will have to converted to “if-then” form)

2. For all matches, losing side consists of at least 1 ducks in the batting innings.

3. For all innings, if strike rate of player is above 200.0 then he has hit more sixes than fours.

(more to be added).

Part 3 and Part 4:

Already sent out in draft 1.

Grading:

20 marks for each part => 20 * 4 = 80 marks