Natural Language Processing Instructions for Students

Semester Examination

narayana.anantharaman@gmail.com

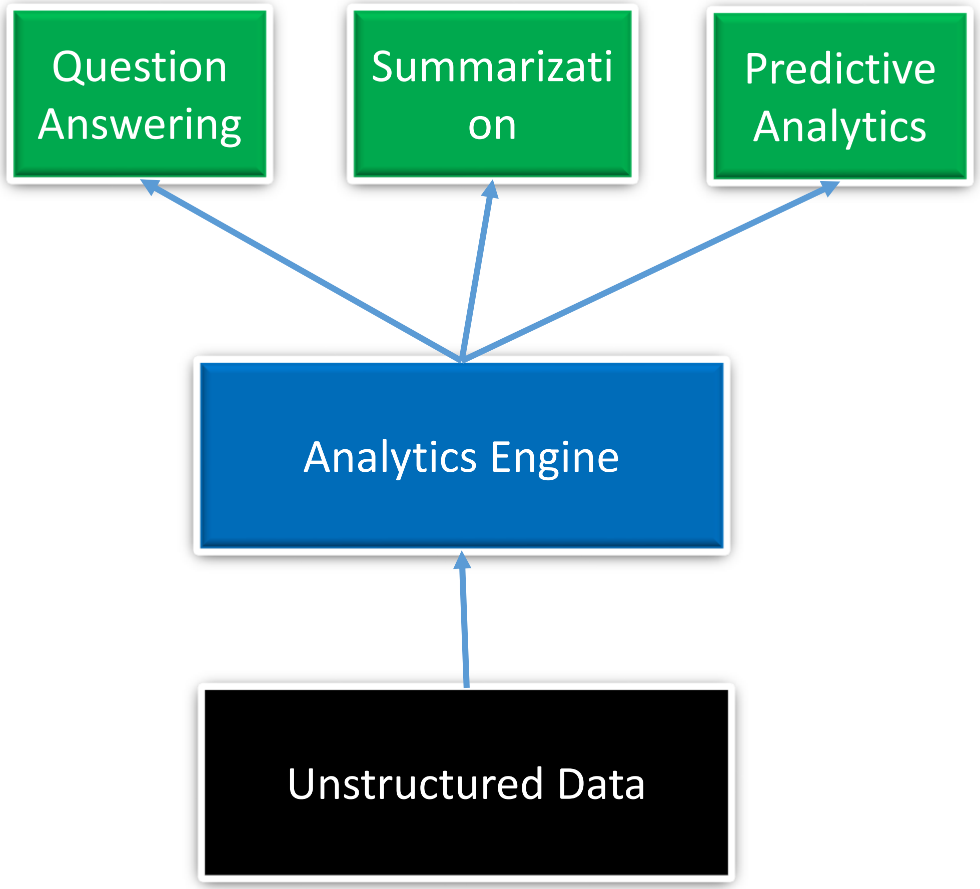
21st Dec 2015 to 23rd Dec 2015

Disclaimer

- The HTML files provided to you were obtained from web pages that are public and using a manual process. Though these data are public, they may be bounded by copyrights.
- Use of these data are strictly for academic purpose and no commercial usage is intended.
- Students are required to delete ALL the HTML data files once the assessment is completed.

Product Goals

- Our goal is to build a text analytics product in the domain of cricket matches.
  - E.g The system should be able to answer a question like: “What is the impact of Sachin’s wicket in this match?”
  - System should be able to write a summary of the match: “India were chasing well till the wicket of Sachin that fell at the wrong time”
- The system extracts valuable analytics from the unstructured text and creates a text analytics engine
- The system supports interesting applications on top of this text analytics engine that include:
  - Question Answering
  - Summarization



Broad Plan

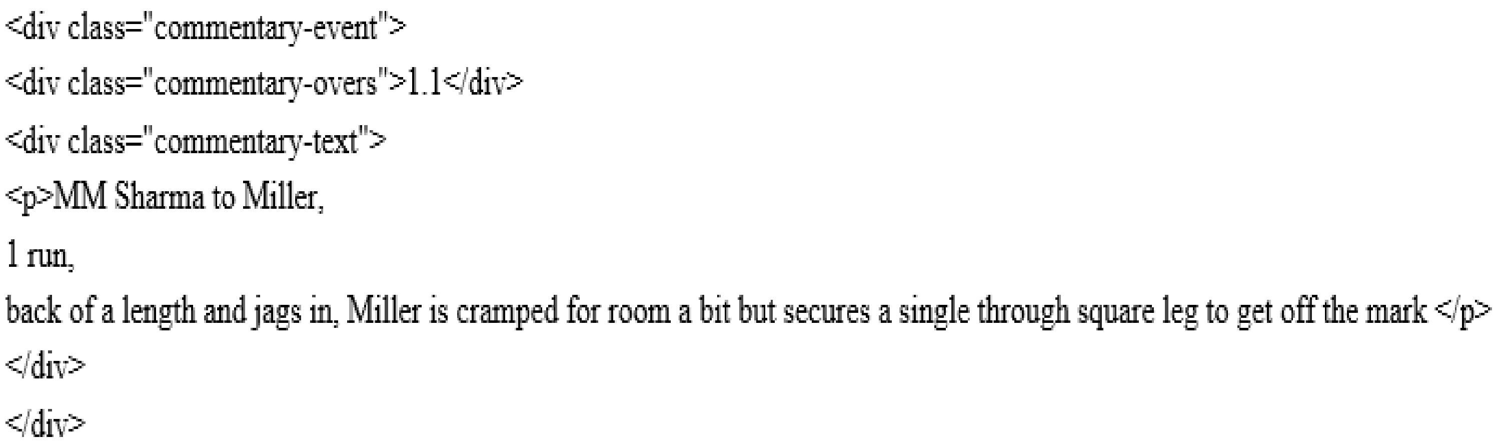
- Focus on getting the analytics engine right during the first 2 days
- Implement one or two simple applications on day 3

Grading Policy

- The goal of this final exam pattern is both to assess the performance of the student in this subject as well as to encourage creativity. To accomplish this dual objective:
    - We have specified mandatory checkpoints that will be assessed for grading
    - The creativity opportunities will be identified in the question paper (this slide deck) and they are meant for bonus score

Task 1 – Extract commentary

- Given: HTML source file of cricket comments of 1 day match, extract the raw text of commentary
- Each file corresponds to 1 innings of <= 50 overs. Two files constitute a match
- Write the text output in to files named as: (original file name).txt. That is, if the original file is named x.html you should generate the output to x.txt
- Optionally, you are required to preserve any meta data for later use.
  - You can write this to a separate JSON or XML file



```
<div class="commentary-event">
<div class="commentary-overs">1.1</div>
<div class="commentary-text">
<p>MM Sharma to Miller,
1 run,
back of a length and jags in, Miller is cramped for room a bit but secures a single through square leg to get off the mark </p>
</div>
</div>
```

<u>Expected Output for the above HTML snippet</u>

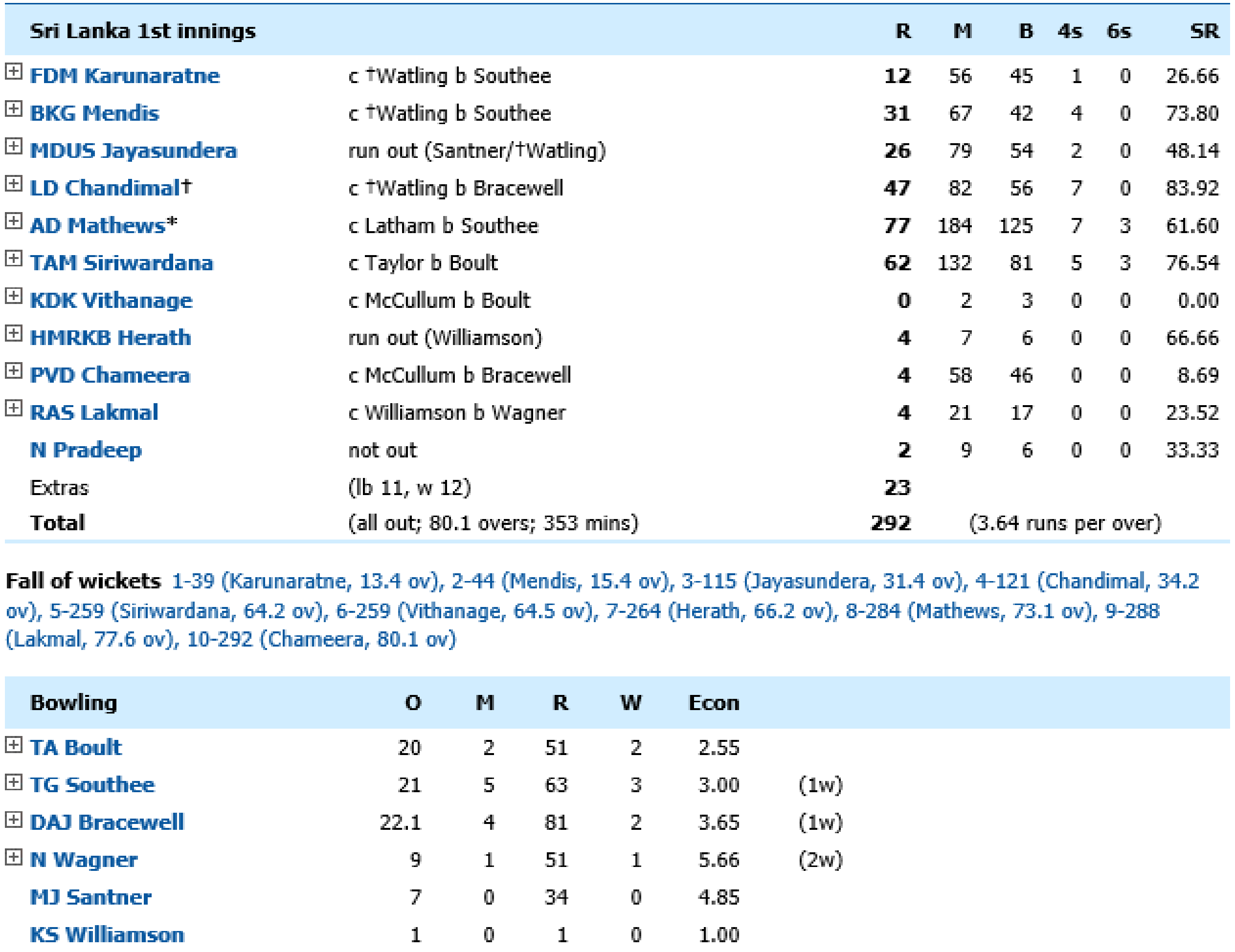
MM Sharma to Miller, 1 run, back of a length and jags in, Miller is cramped for room a bit but secures a single through square leg to get off the mark

Check Point # 1: Demo of text files generated, optional JSON files

Task 2 – Generate the Score board

- Given the text files corresponding to a match (you will have 2 files per match), generate the scoreboard for the match by using only text processing.
  - You need to produce the necessary data structures in JSON
  - You are free to design your schema
  - Visualization not needed but optional

- You can implement rule based or ML based methods as you wish. For example: you can create a rule template like <Bowler> to <Batsman> and match this against the commentary to identify the bowler and batsman involved in that snippet.
  - **Consider the snippet**: MM Sharma to Miller, 1 run, back of a length and jags in, Miller is cramped for room a bit but secures a single through square leg to get off the mark
  - **The rule template: <Bowler> to <Batsman> will identify MM Sharma and Miller as bowler and batsman respectively.**



Sri Lanka 1st innings		**R**	**M**	**B**	**4s**	**6s**	**SR**
⊞ **FDM Karunaratne**	c †Watling b Southee	**12**	56	45	1	0	26.66
⊞ **BKG Mendis**	c †Watling b Southee	**31**	67	42	4	0	73.80
⊞ **MDUS Jayasundera**	run out (Santner/†Watling)	**26**	79	54	2	0	48.14
⊞ **LD Chandimal†**	c †Watling b Bracewell	**47**	82	56	7	0	83.92
⊞ **AD Mathews***	c Latham b Southee	**77**	184	125	7	3	61.60
⊞ **TAM Siriwardana**	c Taylor b Boult	**62**	132	81	5	3	76.54
⊞ **KDK Vithanage**	c McCullum b Boult	**0**	2	3	0	0	0.00
⊞ **HMRKB Herath**	run out (Williamson)	**4**	7	6	0	0	66.66
⊞ **PVD Chameera**	c McCullum b Bracewell	**4**	58	46	0	0	8.69
⊞ **RAS Lakmal**	c Williamson b Wagner	**4**	21	17	0	0	23.52
N Pradeep	not out	**2**	9	6	0	0	33.33
Extras	(lb 11, w 12)	**23**					
Total	(all out; 80.1 overs; 353 mins)	**292**	(3.64 runs per over)				

Fall of wickets 1-39 (Karunaratne, 13.4 ov), 2-44 (Mendis, 15.4 ov), 3-115 (Jayasundera, 31.4 ov), 4-121 (Chandimal, 34.2 ov), 5-259 (Siriwardana, 64.2 ov), 6-259 (Vithanage, 64.5 ov), 7-264 (Herath, 66.2 ov), 8-284 (Mathews, 73.1 ov), 9-288 (Lakmal, 77.6 ov), 10-292 (Chameera, 80.1 ov)

Bowling	**O**	**M**	**R**	**W**	**Econ**	
⊞ **TA Boult**	20	2	51	2	2.55	
⊞ **TG Southee**	21	5	63	3	3.00	(1w)
⊞ **DAJ Bracewell**	22.1	4	81	2	3.65	(1w)
⊞ **N Wagner**	9	1	51	1	5.66	(2w)
MJ Santner	7	0	34	0	4.85	
KS Williamson	1	0	1	0	1.00	

Check Point # 2: Demo of scoreboard data structure including scoreboard, fall of wickets and bowling analysis

Definitions needed for next stage of work

- Discourse
  - A snippet of commentary is considered as one discourse.
    - E.g. For an over that has 6 balls (assuming there are no wides or no-balls) there will be 6 discourses.
- Events
  - Each discourse will have one or more events.
    - For example: A catch dropped is an event and a run scored off a dropped catch is another event. In this example both events are true while other events (such as batsman getting out due to catch) are false.
  - The detailed list and definition of events are presented in another slide.
  - **Caution**: You should stick to these events and not modify any of these as every student will create datasets as per this.

Events

- Events represent the vital aspects of a discourse as a data structure.
- Once we are able to transform a discourse text in to an event data structure, we can use it for many purposes, such as:
  - Text analytics
  - Question answering
  - Summarization
- Each event may have one or more parameters.
  - E.g. A OUT event may have the following:

    { Event_type: “OUT”

    Parameters: {Batsman: Sachin, Bowler: Wasim Akram, how_out: caught, fielder: Shahid Afridi, fielding_position: mid on, catch_difficulty: very_tough, ball_num: 3, over_num: 7, fow_run: 22, fow_wick: 1},

    Impact: HUGE,

    Explanation_for_impact: India chasing > 300

    }

Application of Events

- Text analytics that can answer questions like:
  - What is the quality of Sachin’s batting in this match?
    - Can be computed as a function of number of runs scored, nature of pitch, quality of the bowler, number of bouncers, number of Yorkers etc
  - How many bouncers were bowled by Wasim Akram?
  - How many boundaries Sachin scored through leg side?
  - How did he reach his century?
  - How many times he was beaten?
  - Which bowler troubled him the most?
- We can build a highlights package by going through the events and selecting most important events to include in the highlights

Events List (Mandatory)

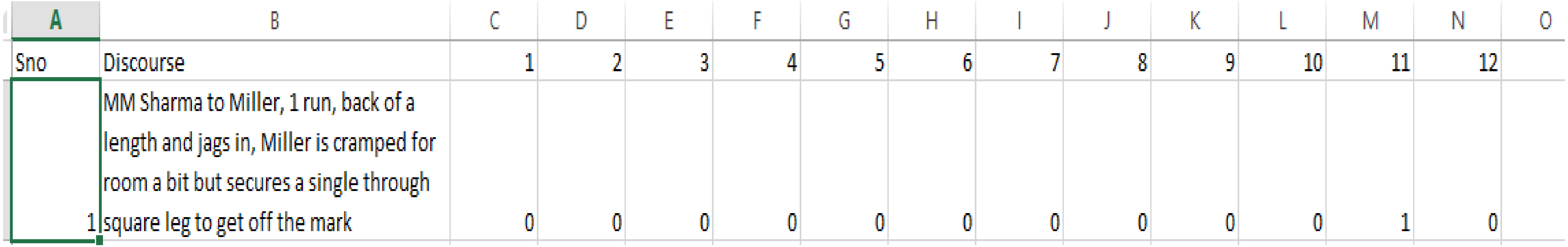
1. Defended
2. Left alone
3. Beaten
4. Edged
5. Caught
6. Runout
7. Stumped
8. Bowled
9. LBW
10. Boundary_scored_by_batsman
11. Runs_by_batsman
12. Boundary_scored_extras
13. Runs_by_extras
14. Catch_dropped
15. Stumping_missed
16. Runout_missed
17. Bouncer
18. Yorker
19. Overthrow
20. great_save
21. poor_fielding
22. Free hit

What are we going to do with events?

- First we need to define the events such that a rich database of text analytics can be constructed. This involves defining the event_type and defining the parameters for that type.
- We then need to extract the events from the discourse. This is a process of Information Extraction where the discourse is an unstructured text and the output is the event data structure
- Given the discourse, this can be viewed in to 2 parts: Extracting the event_type and extracting the parameters for that type.
- In day 1 and part of day 2 morning we will focus only on detecting the event_type
- As a first step we need to tag each discourse in to a target bit vector of events
- **Caution**: You are required to use a ML approach for the event_type detection, though hard coded rules also might work. You are NOT allowed to use rule based approach for this task because we should assume different commentators would follow different styles.

Instructions for Dataset Creation to detect events

- You will be provided a template (screen shot as below) and you are required to construct the dataset strictly as per the template. Your data will be merged with the data from other students and so any mismatch will make the effort unusable.



A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Sno	Discourse	1	2	3	4	5	6	7	8	9	10	11	12	
1	MM Sharma to Miller, 1 run, back of a length and jags in, Miller is cramped for room a bit but secures a single through square leg to get off the mark	0	0	0	0	0	0	0	0	0	0	1	0	

Task 3 – Create the dataset

- Use the template given and tag each discourse. The target vector is a binary of 21 elements. The bit position corresponds to event_type (bit_0 is defended event, bit_21 is free_hit event)

Check Point # 3: Your dataset will be inspected by the faculty

Task 4: Build the classifier

- Develop a classifier that predicts the output events given the discourse. You are required to train the model using 75% of the dataset and test it with 25% remaining data.
- Some suggestions:
  - You can use any machine learning classifier for this problem. A simple MaxEnt would usually be adequate, because it is possible to write accurate feature functions for this problem.
  - Note that the output layer is logistic as more than one event may be true in the given discourse.
  - You may also try converting the discourse in to a fixed sized feature vector and train a feed forward neural network.
  - RNN might be an overkill.

Check Point # 4: Demo of classifier and reporting the classification accuracy

Extracting the parameters for the events

- The parameter extraction depends on the richness of applications that we need to support
- This can best be arrived at by coming up with sample outputs

Task 5 – Corpus Creation

- If your project group number is an odd number do the following:
  - Each student will come with 5 questions pertaining to a one day international match.
  - Each question should be one line.
  - Place your questions in a text file and name it as: q_<your_usn_number>.txt
  - Note: Your file should only have questions and no extra elements like serial number etc and should be a plain text.
- If your project group number is an even number do the following:
  - Write a simple text summary per innings of a match. Thus you are required to write 2 summaries per match.
  - Place the summaries in the text file and name it as: s_<your_usn_number>.txt

Check Point # 5: Datasets submission