**Automated Cricket News Generation in Sri Lankan Style using Natural Language Generation**

1. **Introduction:**

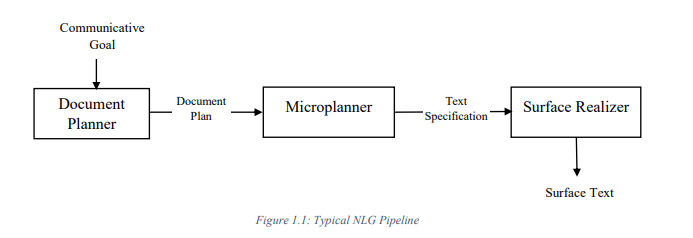
The section emphasizes the growing challenge of handling vast data and the role of Natural Language Generation (NLG) in making this data comprehensible. It explores the importance of automating document generation across various domains. In the context of journalism, especially sports journalism, NLG is recognized for its potential to streamline the editorial process. Cricket is chosen as the domain for research due to its popularity in South Asia, standardized input from cricket match scorecards, and the availability of domain knowledge. The research aims to generate journalistic pieces on cricket matches, investigating NLG techniques, template-based approaches, and their adaptability to different match situations. The system's output could potentially be used for automated narration through Text-to-Speech (TTS) systems. The focus on Twenty20 (T20) cricket matches adds novelty, addressing a gap in existing research on cricket summaries.

1. **Problem Statement:**

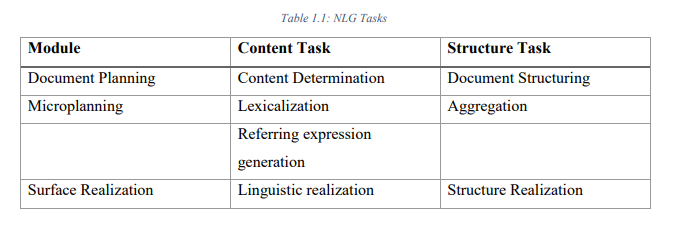
The problem identified revolves around the challenges faced by journalists in the rapidly evolving technological landscape. The need for immediate news reporting, especially in the context of multiple concurrent cricket matches, requires significant time and effort from journalists, limiting their capacity for in-depth reporting. Automation is proposed as a solution to enhance efficiency and professionalism. The automated process is expected to reduce journalists' workload, allowing more focus on in-depth reporting. While acknowledging the limitations of automated journalism, particularly in achieving qualitative and user-friendly content, the paper emphasizes the need for adapting templates to match-specific situations, addressing the domain independence of existing template-based systems. The lack of research in automated sports journalism, specifically in generating cricket news in the Sri Lankan style, is highlighted as a gap to be addressed.

1. **Solution:**

The paper proposes a Natural Language Generation (NLG) solution to address the challenges in cricket news reporting. It distinguishes between Real or Standard NLG, focusing on theoretical foundations, and Template-based NLG, which directly maps non-linguistic input to linguistic surface structure. The chosen solution opts for Template-based NLG. For a practical NLG system, a well-defined architecture is essential, and the paper advocates for a decomposed structure with distinct, easily integrated modules. The selected high-level architecture is a Pipeline architecture, which is depicted in Figure 1.1.



There are six (6) NLG tasks which are proposed by Reiter and Dale [7], attached with the pipeline architecture. They are Content determination, Text structuring, Sentence aggregation, Lexicalization, Referring expression generation, Linguistic Realization. These tasks are performed with each of the above-mentioned module in the pipeline architecture.



1. **Objectives:**

The primary objective of the research is to automatically generate cricket news from structured databases. The specific intentions aligned with this main objective include:

* **Generate Journalistic Pieces in Sri Lankan Style:** Develop news articles using the Sri Lankan style of English cricket news reporting.
* **Achieve Variation in News Summary:** Create diverse news summaries based on different situations within a cricket match.
* **Understand and Restructure Score Card Information:** Comprehend structured information from cricket match scorecards and restructure it into a specific format.
* **Generate Grammatically and Syntactically Correct Text:** Ensure the generation of text that is grammatically and syntactically correct while maintaining meaningful content within the domain.
* **Compare and Contrast with Reference Text:** Evaluate the output text by comparing and contrasting it with a reference text, assessing its accuracy and quality.

**5. Research Methodology:**

The research methodology focuses on generating journalistic pieces on cricket matches using a template-based Natural Language Generation (NLG) system, primarily relying on 20-20 cricket match scorecards from Cricinfo. The approach involves extracting key semantic concepts from the scorecards to create a natural language summary.

**Challenges in News Generation:**

The presented templated scorecard for a 20-20 match illustrates the complexity of generating news with rich language and user-friendliness. Some details, like Rohith Sharma's T20 hundred and Hardik Pandaya's career-best wickets, are not directly available in the scorecard and require additional input to the system. The goal is to produce a natural language summary that incorporates the provided input and extracts key semantic concepts.

**Sri Lankan Context of Sports Journalism:**

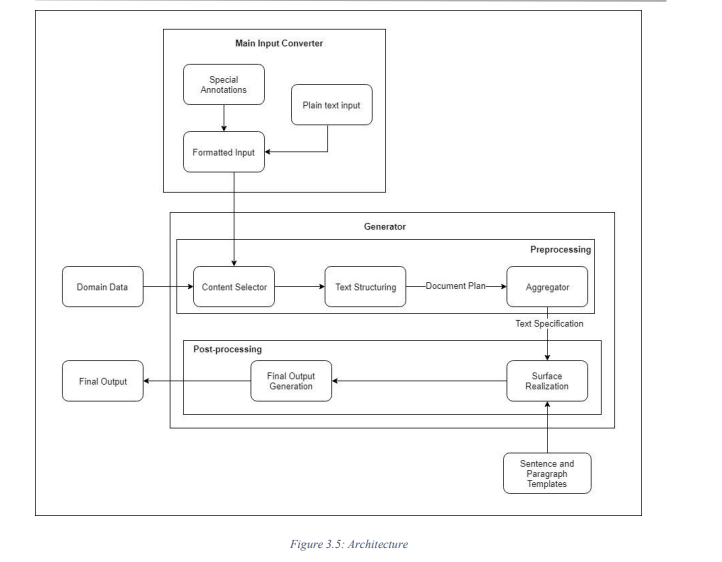
The research specifically targets generating cricket news in the context of Sri Lankan sports journalism. While automated journalism in Sri Lanka is an unexplored area, there is a distinct difference in the style of Sri Lankan cricket news compared to internationally recognized sports websites like Cricinfo. Sri Lankan cricket news typically presents match summaries without in-depth reporting and expert judgments.

**Example Comparison:**

An example comparison between a Sri Lankan cricket news summary from the Daily Mirror and an internationally recognized sports website like Cricinfo emphasizes the differences in reporting styles. The Sri Lankan context tends to focus more on match summaries without extensive analysis.

**System Architecture:**

The proposed system employs the standard pipeline architecture for Natural Language Generation (NLG) systems. This architecture is chosen for its low dependency between phases, with each component having well-defined tasks to transform data in a specific manner. The system architecture includes six NLG tasks: Content determination, Text structuring, Sentence aggregation, Lexicalization, Referring expression generation, and Linguistic realization.



**Input Processing:**

* The main input is a 20-20 cricket match scorecard from Cricinfo, converted to a JSON tree for the text generator module.
* Domain data, paragraph and sentence templates serve as additional inputs to the generator module.

**Main Input Converter:**

* Converts the plain text scorecard into a structured JSON tree.
* The JSON tree includes sections for game overview, teams, scores, and background information.

**Content Selector:**

* Rule-based approach for content selection based on predefined criteria.
* The trainable approach can be explored in the future for comparison.

**Text Structuring:**

* Organizes selected content into sections (e.g., introduction, result, overview) based on predefined rules.
* Content selection and text structuring are performed together.

**Aggregator:**

* Determines which items can be aggregated into a single sentence, discarding duplicates.
* Groups similar items based on concepts.

**Surface Realization:**

* Takes output from the aggregator, linguistic functions, grammatical functions, and templates.
* Linguistic functions convert numeric or date formats to text, while grammatical functions maintain coherence and concordance.
* Utilizes sentence and paragraph templates.

**Final Output Generation:**

* Performs minor validation checks, making relevant changes if needed.
* Output includes the final summary generated based on the given domain.

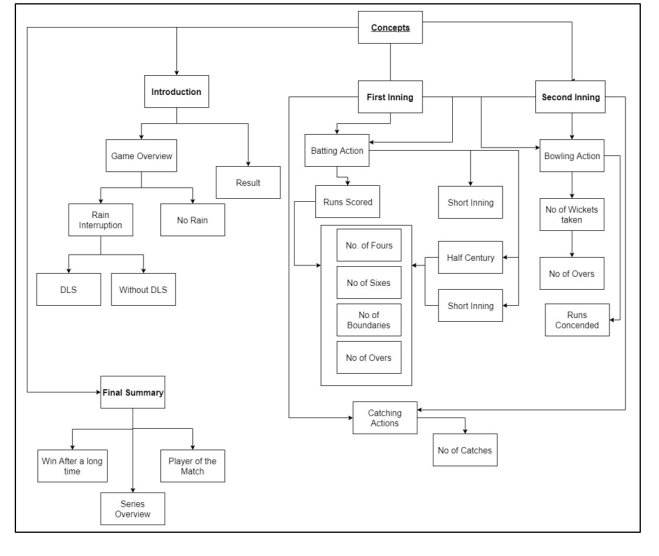
1. **Research Design:**

**Data Collection:**

* The primary input, the scorecard, is sourced from Cricinfo in plain text format.
* Additional inputs include domain data (team players and team information) obtained from Cricinfo and stored in a database.
* Templates are designed based on DailyMirror cricket news [28].

**Templates Design:**

* Two types of templates are designed: Sentence Templates and Paragraph Templates.
* Templates cover basic match information, including batting, bowling, and fielding actions.
* Key semantic concepts are incorporated into templates.

****

**Background Information:**

* Injected into the system from database values or runtime inputs.
* Includes details such as match date and location, injured players, rain interruptions, use of Duckworth-Lewis-Stern method, and whether the winning team had a long-awaited victory.

**Template Storage:**

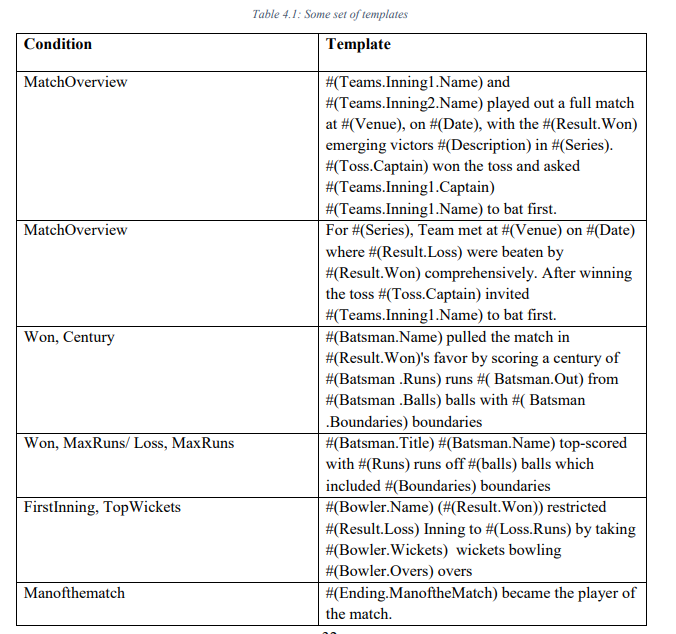
* Templates are saved in an Excel file.
* Selection of templates is based on keys (annotations) associated with each template.
* Each template can be based on one or more annotations, allowing flexibility.
* Templates contain gaps assigned variable names; assigning values to variables occurs in the generator module, enabling output variation.

**Language Extension:**

* Templates are fixed, facilitating system extension to different languages.
* Conversion to other languages (e.g., Sinhala) is achieved by translating template content while retaining variable names.

**Example Templates:**

* Several templates are defined for different match conditions, providing flexibility and variety in generated outputs.

****