European Journal of Computer Science and Information Technology
Vol.7, No.6, pp.42-56, December 2019
Published by ECRTD- UK
Print ISSN: 2054-0957 (Print), Online ISSN: 2054-0965 (Online)

AUTOMATED CRICKET NEWS GENERATION IN SRI LANKAN STYLE USING NATURAL LANGUAGE GENERATION

D. Y. Gunasiri, K. L. Jayaratne
University of Colombo School of Computing

ABSTRACT: *Cricket is one of the most followed sports in South Asia. There is a wide requirement for news to get produced within a short period after the cricket match. Within Sri Lankan publishing this news article is a manual process and needs journalists with domain and language competencies. Most of the time it is not possible for sport journalists to focus on in-depth reporting due to time and cost constraints. Therefore, automated process would be efficient and cost effective. As a solution, an Automatic Cricket News Generation System is presented through this research and how a template based natural language generation is utilized in implementing such system and its suitability are demonstrated. The scorecard is used as the main input for this system which could generate a journalistic summary in Sri Lankan style and this implementation is based on pipeline approach in NLG. Furthermore, it also focuses on the variation of the output generated via the system which is not much typically be in use in template-based NLG systems.*

KEYWORDS: NLG, Template-based NLG, Pipeline approach in NLG

INTRODUCTION

With the vast development of technology, there is a significant increase in data collection. But some expertise knowledge is needed to manipulate and interpret those data in a human understandable form. Natural Language Generation (NLG) comes into play in such scenarios. NLG is a sub task of Natural Language Processing (NLP) where it uses techniques from Artificial Intelligence (AI) and Computational Linguistics (CL) to generate human understandable text automatically [1].

NLG is getting popular in the context of journalism since the increase in availability of and access to data on a scale that prevents journalists from handling and using it for news reporting. The "robot journalism" or automated journalism [2], has reduced the variable cost for the journalism to the zero level. In the domain of sports journalism, due to the abundant availability of information on databases about cricket matches, journalists might find it difficult to organize those data to produce news. If this editorial process is automated, journalists can spend time on in-depth reporting thus reducing time spent on large amount of data including statistics and numbers.

Cricket is one of the most popular sport in South Asian countries and it is getting popular among the other countries in the world too. Therefore, Cricket was chosen as the domain for sports journalism in this research. The main objective of this research is to generate journalistic piece on cricket matches by investigating different NLG techniques and the suitability of template-based approach for generating news on cricket matches. The key aspect of this research is to have different template sentences

which are changing according to the situation of the match and thereby increasing the domain dependency. This system would produce a natural language output of text from structured data in a non linguistical format. Therefore, cricket news piece for a given match will be generated using the match score card.

There are several forms of cricket matches recognized by International Cricket Council (ICC). Since Twenty20 (T20) cricket matches are getting popular day by day and it is noted that there are no researches were done in producing cricket summaries for T20 cricket matches. Therefore, it was decided to generate news for Twenty20 cricket matches and the rules together with other related information that would be unique to T20 matches should also be considered when developing the NLG system.

BACKGROUND

Over the last few years there has been a rapid development in technology which leads to an enormous change in many of the expectations of the community. For example, if an event happened a day before, people expect to see a full article on it the next day itself. Most of the time people used to gather few points in their mind and try to deliver these points as an article. Therefore, automating this process may help to make it more efficient and professional. Today, there are many cricket matches that take place at the same time and it is important to report news on all these matches. Reporting news requires significant amount of time and effort of the journalists and they need more domain knowledge as well. If this editorial process is automated, more working time of journalists can be reduced considerably while they can spend more on in depth reporting.

Journalism is not just description; it is an event driven story. Journalists need to be highly skilled in recognizing, creating, organizing and communicating stories in natural language. It is difficult to get the quality of journalism through an automated process since there are limitations and restrictions in the automated process of news generation compared to manual journalism. Most of the automated journalistic pieces are created through a template with a shorter description. However, extending the journalism from a short descriptive report to a qualitative journalistic piece is challenging in since the absence of required data may also lead to less user friendliness in the news. These templates used in automated sport news reporting, are more domain independent and general purpose. This needs to be addressed by this research and it would discuss how the user-friendliness is achieved by changing the template sentences according to the situation of the match.

Many researches were done for automated journalism, but the area of sports is not covered much. Furthermore, there is a need for a research to generate cricket news automatically according to the Sri Lankan style and it is also not covered by the current researches which were done related to sports journalism.

LITERATURE REVIEW

A. Natural Language Generation

Natural Language Generation (NLG) is a subfield of Artificial Intelligence and Computational Linguistics. It focuses on generating understandable texts in English or any other language by typically starting from non-linguistic representation of information as input [4]. NLG can be regarded as an emerging technology since many real-world applications have been built based on this technology. From research perspective, NLG is a subfield of Natural Language Processing (NLP). It is possible to distinguish NLG systems as two types depending on the input. They are D2T (Data to Text) and T2T (Text to Text) [5].

B. NLG Tasks

There are six NLG tasks identified in most of the NLG systems when converting the input to a human readable form of text [5].

- Content Determination
  At this stage it decides on the information to mention on the text. It filters out the most important information to be shown in the final output text.
- Text Structuring
  This will determine the order of information in the text.
- Sentence Aggregation
  Every message need not be showed in separate sentences. By combining multiple messages into single sentence, it would be more fluid and readable.
- Lexicalization
  Lexicalization is finding the correct words and phrases to express the information.
- Referring Expression Generation
  This action would select the words and phrases to identify domain entities. This would avoid ambiguity.
- Linguistic Realization
  This task involves ordering constituents of a sentence, as well as generating the right morphological forms. Here it would form a well-formed sentence.

C. Template Based Approach

Template based systems are the natural language generation systems which maps non-linguistic input directly to linguistic surface structure [6]. This linguistic structure has gaps and well-formed text would be an outcome when all those gaps are being filled with linguistic structure that do not contain gaps. According to Busemann and Horacek, there are two orthogonal methods which help for the efficient development in NLG systems, such as 1) general, reusable, linguistically motivated surface realization components 2) simple task-oriented template-based systems [7]. It is stated that use of these two methods are also limited. Because, using surface realization component is problematic since domain oriented and linguistically motivated ontologies are different.

At the same time, it is observed that existing template-based techniques are inflexible [7].

Natural Language Test Case Templates (NLTCT) is a template-based NLG methodology [8]. Through this it is able to achieve some kind of a flexibility through a template-based NLG method. It generates NL description from each Z test case where it has parametrized operations. It can insert translation rules for these. Although NLTCT is said to be inflexible but one of the limitations identified here is the domain independence. At the same time, it is difficult to maintain several test cases when the domain becomes larger.

Reiter and Dale had claimed that template-based system have been substandard in respect to maintainability, output quality, variation and well-foundedness [9]. They are difficult to maintain, update as well. According to author's opinion, these are related to pure template-based systems, if it can maintain a variation in templates many of these drawbacks can be avoided.

There are ways that limited variation can be achieved through template-based systems. They are using parameterized templates and Template Specialization Hierarchies [6]. Parameterization is a flexible way to guide and control the text generation process according to linguistic preferences such as styles, rhetorical structures. Here it dynamically sorts out the application of templates according to parameters. XTRAGEN [10] application is one of the examples which uses this method.

In template specialization hierarchies, decision tree traversal is taking place according to applicability conditions. EXEMPLAR [11] is one of the best examples uses this method which it finds the most specific exemplar for the current context. Using these above-mentioned methods, advantages such as efficiency, simplified system architecture, full control over output and reduced demand on knowledge acquisition and representation can be achieved through a template-based NLG system.

D. Applications of NLG in Related to Journalism

Automated journalism is relatively new and technologically innovative field, but this area is touched through several researches. Los Angeles Times was the first newspaper to report the earthquake happened in 2014 close to Beverly Hills, California. This was reported within 3 minutes and it was automatically generated by a 'robot journalist' which converts the input parameters to a pre-defined template [12]. This was a data-to-text generation system. Automated journalism needs data in machine readable format like spreadsheet or the data should be converted to a machine-readable format through machine learning or any other technology. Automated journalism cannot be used for the domains with lack of data and when quality of data is poor. Additionally, the six requirements [13] identified in automation of journalistic process are transparency, accuracy, modifiability and transferability of the system, fluency of output, data availability and topicality of the news.

E. Automated Journalism Related to Sports Domain

Sports is one of the ideal domains for automated journalism because of rich data availability and input formation and style in news reporting. Several sports are being researched in sports automated journalism and some of them are baseball, soccer, football and cricket.

GoalGetter is a data to speech system which generates spoken summaries of football matches in Dutch [14]. In order to generate, it takes in the information in tabular format and this is similar to a system called STREAK [15] which generates summaries of basketball instead football. STREAK generates summaries using a revision-based approach to summarization. In the first pass, it generates draft with fixed information like who won, lost and in the second pass, opportunistically add in information, as allowed by the form of the existing text. The difference of GoalGetter from STREAK is that STREAK generates a written output while GoalGetter produces a spoken output.

The GameRecapper [16] system is a template-based system that generates Portuguese summaries of football matches from structure input data such as game sheets taken from www.zerozero.pt website. This data is converted to JSON tree and the implementation of GameRecapper is also based on GoalGetter system. GameRecapper generation module is basically the algorithm that creates the news. It utilizes the domain data, linguistic function, grammatical function and a collection of sentence templates. As in GoalGetter, domain data would provide the additional data to produce more variations in the output text while grammatical function ensures the coherence in the text. Linguistic function would convert the numeric values to words since this domain contains more numbers in the input data. The templates are divided into groups according to several types of events in the game such as first goal, game end etc. or different characteristics of the game such as home team became the winners etc.

Multilingual Sport Summary Generator investigates the suitability of template-based system to generate multilingual sport summaries. The system generates English and Bangla Cricket summaries using score card, which is very much similar to the requirement in this proposed research. They have used pipeline architecture since it has low dependency while maintaining clear separation between each step and allows the extendibility of modules. Input of the system is a cricket match score card taken from cricinfo [3]. Additionally, to the input, the system is fed with custom annotations which includes the background information which cannot be deduced by in the input score card itself. In this system contains six custom tags which can be used to describe batting conditions and bowling conditions [17].

In Tamil Cricket Summary Generator summaries are generated in Tamil using cricket match score card. This system makes use of two parameters namely, interestingness and humanness. The cricket summary generator has following four components namely, data gathering and modelling module, data mining and data analytics module, summary generator and Evaluator [18].

RESEARCH DESIGN

The architecture of the proposed system is based on standard pipeline architecture of NLG systems. The reason behind using the pipeline architecture is that the phases of the architecture has low dependency and each component has well defined tasks that would transform the data in a specific manner. One of the disadvantages of this architecture is that it is one way unless it is customized. One such example would be that is not easy to change the content selected at the content determination stage at latter stages. Nevertheless, the generation is one-way pass and it would not need to go back and change what has done.

Figure 1 depicts the main architecture design for the proposed system in this research. According to the architecture, the generation module is mainly based on the traditional pipeline architecture. As shown, the input generator works as a separate module since the text generation process only interests on the final input. This final input needs to be in a proper standard structure and in this case, it would be a json format. This json tree would characterize the basic information about a selected match.

The domain data, paragraph and sentence templates are another source of input to generator module. This domain data module contains different information that helps to generate the final output. For example, it would contain the information about the teams. The coach of the team, the players who are not playing for the given match from the team, fresh players of the team are some of the information about teams.

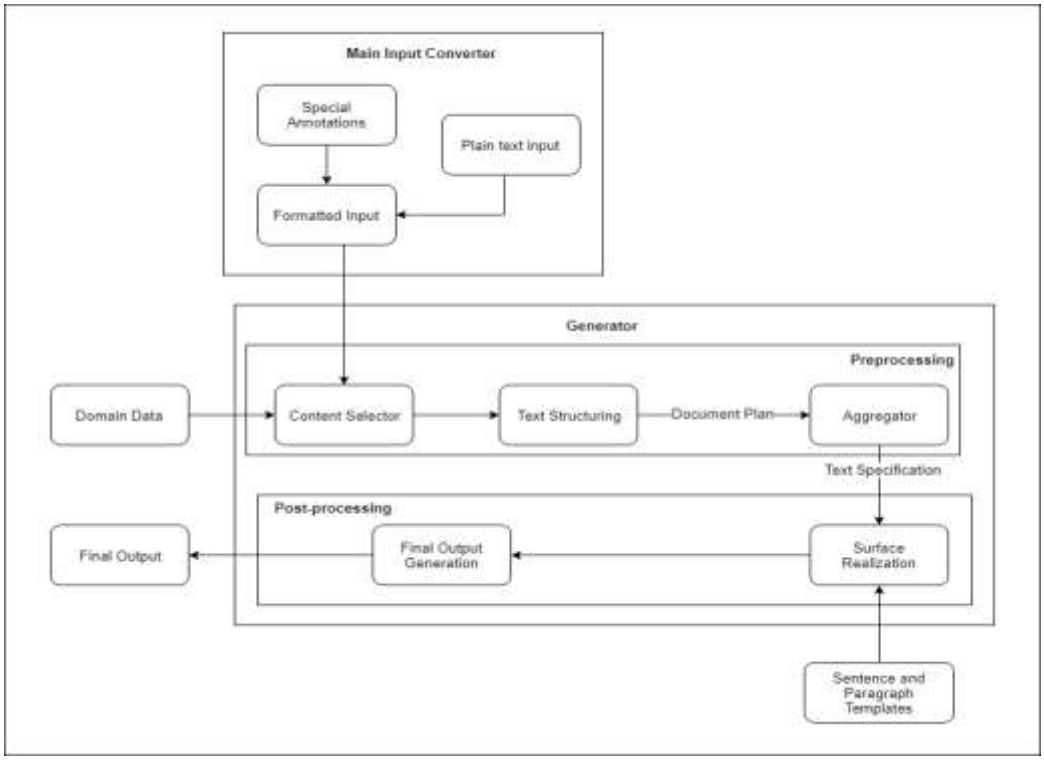

Figure 1: System Architecture

Sentence and paragraph templates are one of the main inputs to generator module. These are incomplete sentences or paragraphs with open slots for variables. There are several similar templates which would be used according to the instance. Choosing the correct template is a task for the generator module. Variation in the final output depends on the sentence templates choices that are made. Templates are divided in to groups according to some criteria such as where to be used in the final output. For example, sentence templates for the introduction would be grouped together.

A. Main Input Converter

As mentioned above score card for the relevant cricket match would be the main input to the system. But this score card is in the plain text format. The plain text is converted to JSON tree by a parser while an algorithm will be used to make this task fulfilled. This JSON tree acts as the main input to the text generator module and this JSON tree must be properly structured so that it would be easier to be utilized by the generator. The subsections of the JSON tree would be the game overview with the game results and teams played, scores by the two teams, background information such as location of the game etc.

Corresponding to each player there will be a separate node and each of these nodes consists of leaf nodes which contains the performance during the match. Special annotations will also be attached with each player node if there is anything special to be included. The special annotation tags will contain the specific background information on each player. For an example, with this player node system can deduce that the player got out by playing a poor shot. Background information needed for the input module will be provided through a database. These custom tags would make the final output text more meaningful. A part of json output which is generated through the input module for a match is given below in Figure 2.

```
"Batsman4":{
        "Name":"Eoin Morgan",
        "Catch":"Dhoni",
        "Ball":"Hardik Pandya",
        "Out":"Out",
        "Runs":"6",
        "Balls":"9",
        "Fours":"0",
        "Sixs":"0"
        "Role": "Batsman",
        "More": "got out by playing a poor shot"
    }
```


Figure 2: Player Node

B. Content Selector

The content selector module would get the input and would manipulate according to some criteria given. There are several approaches for content selection. Some of them

are rule based content selection and trainable content selection [1]. To use the learning method there should be a significant amount of input data. Therefore, in this proposed system, the rule-based approach would be used. According to some predefined rules the formatted input to the generator module will be manipulated and organized. The content selection is responsible for picking items that would only be needed for the realization.

The formatted input contains all the information about every batsman and bowler, but all these data may not be needed for the realization. Therefore, choosing the batsmen and bowlers who have well performed or should be included in the final output will be selected by the content selection module based on some rules and fuzzy logic. Some of the rules which are used for the content selection in batting perspective are players with maximum runs scored, players who have scored half century, players who have scored century.

C. Text Structuring

After the content selection is performed, text structuring module is responsible to order these items that were chosen. According to sub sections of the final output such as introduction, result, overview etc., the text selected will be organized. Similar concept will be grouped together according to a rule-based method. The rules are defined with weights and according to weightage the items will be ordered in each batting, bowling and catching sections. Content selection and Text structuring will be performed together.

D. Aggregator

Aggregation is the phase which is accountable on decide on which items to aggregate together from the output taken form text structuring. This is the fourth stage of the data transformation. According to Kalina Bontcheva, values with the same property and domain it is able to perform sematic aggregation. Therefore, such values can be expressed within one sentence [19].

In this research context, it will try to aggregate the similar items based on the order and type. Since the items were ordered in prior modules according to the weightage, there is a possibility to aggregate the consecutive items accordingly and it is not compulsory to perform aggregation. In above what it is mentioned as types are ‘batting’, ‘bowling’ and ‘catching’. If the types are similar in consecutive items, they have the potential for aggregation. There are some predefines rules that conforms the aggregation.

E. Surface Realization

The Surface Realization Module would take the output of the aggregator module together with the linguistic functions, grammatical functions and templates. The linguistic functions would change the numeric or date format to text format. The grammatical functions would keep the coherence and concordance of the text. Singular and plural words and grammatical gender (male and female) words would be handled

here. Templates will be used within this module. Templates are designed based on the key semantic concepts shown in figure 3.

The keys in the templates will be mapped with the type of the news item. From that the suitable template for each situation will be selected. Since the variation of the output is one of the major objectives of the research, there can be multiple templates are provided for the same situation and system would select one template at a time randomly. After selecting one template for a given situation, assigning values to the variables based on the values of the internal representation will be done. This will be done by parsing through the given node to find the correct value to assign for the variable. Table 2 depicts an example sentence template used and how its being realized.

Table 1: Template and Realized Output

Template	Realized output
#(Teams.Inning1.Name) and #(Teams.Inning2.Name) played out a full match at #(Venue), on #(Date), with the #(Result.Won) emerging victors #(Description) in #(Series). #(Toss.Captain) won the toss and asked #(Teams.Inning1.Captain) of #(Teams.Inning1.Name) to bat first.	India and England played out a full match at County Ground, Bristol, on Sunday, July 08, with the India emerging victors by 7 wickets in 3rd T20I, India tour of England, 2018. Virat Kohli won the toss and asked Eoin Morgan of England to bat first.

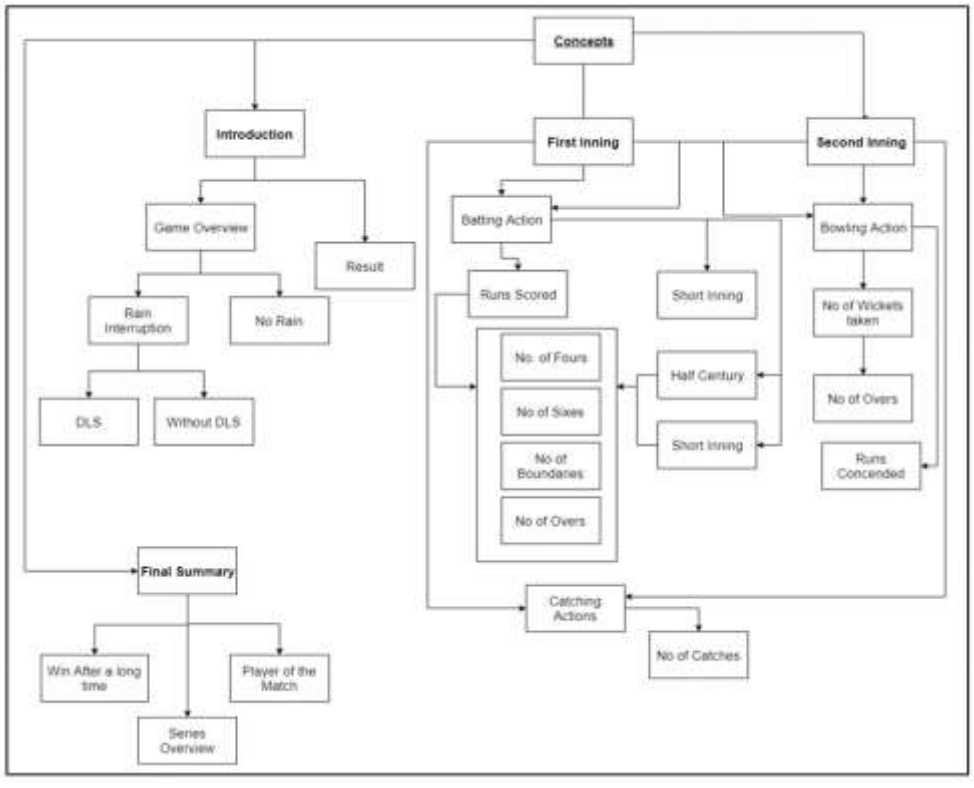

Figure 3: Key Semantic Concept

F. Final Output Generation

In this final output generation, it would get the output of the surface realization module and some minor validation checks will be performed. If these minor validations return false, it would make the relevant changes. Capitalization of the first letter of a sentence is one such validation check. Output of this module will be the final summary generated on the given domain.

EVALUATION AND RESULTS

Generally, NLG evaluation is marked by great deal of variety and it is difficult to compare NLG systems directly. There is no straight forward method or standardized matrix that evaluates this metric. According to Reiter and Dale [2], the success or failure of the NLG system cannot be predicted using the results of a quality evaluation. NLG systems are usually evaluated using user acceptance test. But it is time consuming and there can be factors which would influence the comparison that cannot be considered in Natural Language Generation.

There are two common approaches in evaluation methods of NLG system i.e. Intrinsic Evaluation and Extrinsic Evaluation [22]. The intrinsic evaluation assesses properties of the systems in their own right while extrinsic evaluation assesses the effect of the system on something that is external to it.

There are three basic intrinsic techniques which can be one out of three when it comes to evaluation [1].

1. Assessment by trained assessors of the quality of the system outputs according to different quality criteria, typically using rating scales.
2. Automatic measurements of the degree of similarity between system outputs and reference outputs. E.g.: BLEU and ROOUGE [20]
3. Human assessment of the degree of similarity between system outputs and reference outputs.

For the evaluation of the proposed system, automatic measurements like BLEU or ROUGH will not be used since word by word/phrase by phrase evaluations will not be suitable for a NLG system that would use sentence templates. One of the main objectives of the system is to have variations in the output text. Therefore, it is not feasible to compare the human authored text with the generated output for an exact match.

Here it uses a customized method of evaluation which would compare the system generated text with two reference output texts with help of several parameters. One summary would be an expert written while the other one be an average written summary. The ultimate goal of the evaluation is to achieve the humanness parameter level which is calculated based on the comparison that uses three types of scores.

1. Similarity Score – Ratio of number of nouns present in generated output and one of the reference outputs.
2. Degree of Closeness – The value taken by comparing the generated output text with reference output text under the parameter of number of similar events (Manual)
3. Data Count – This value represents richness in data as a comparison.

A. Similarity Score

An automatic calculation method is used here to represent the ratio of nouns of reference text to generated text. NLTK library functions (ie. Tokenizing, Pos-tagging) will be used to identify the nouns in both the text files. This similarity score will be done twice for both the reference text files such as average written summary and expert summary. This score is calculated based on the following formula.

$$\frac{\text{(No of nouns identified in the reference text) – (No of nouns identified in the generated text)}}{\text{(No. of nouns identified in the generated text)}}$$

This evaluation was done for three generated texts with its corresponding average written summary and expert written summary. The results are given in the Figure 4.

Figure 4: Similarity Score Result

According to results, it is not fair to expect the same pattern of results for every match since the different nouns like synonyms can be used to demonstrate the same situation. However, as the templates are being used to generate the text and these templates are created using the news articles found in Sri Lankan cricket news articles, the similarity score is comparatively low when the generated text is compared with average written summary than comparing it with expert written summary.

B. Degree of Closeness

The score is taken by comparing similar events. Similar events will change from one match to another. Same event may be written using different vocabulary. Manual evaluation is done to get the score. Examples of the events taken into consideration are "Took a hattrick", "poor shot by", "DLS", "won the toss", "agreed to bat first" etc. The result did not show an exact pattern that could identify. Table 2 illustrates the no of events comparison results.

Table 2: No of event comparison

Match	No of events in generated text	No of events in average written summary	No of events in expert written summary
1	11	10	6
2	12	7	26
3	14	9	21

Some of the reasons why the number of events in generated text was less due to the lack of background information. However, some matches show excess information of the results. For example, it may not need to include the number of runs scored over 30 when there are 2 half centuries played by two batsmen. But the generated text includes such information as well due to hard coded rules. Therefore, there is a vague pattern when it comes to number of events.

C. Data Count

This evaluation method would compare the generated text with the reference text to assess the richness in data. The data count will be calculated based on proper nouns. For this calculation, again the NLTK library functions are used. In most of the cases, count in the generated text summary is higher than the average written summary. Table 3 illustrates the data count results.

Table 3: Data count comparison

Match	Generated text	Average written summary	Expert written summary
1	27	14	18
2	33	37	66
3	16	15	48

According to the results for the given three matches, richness in data is in a satisfactory level since the generated text has comparatively similar or higher value for number of proper nouns within the summary. But when it is compared with the expert written summary, it is not easy to find a pattern.

CONCLUSION

This research presented a template-based system which can generate cricket news from a structure format of text data in Sri Lankan style. It is investigated the suitability of templated based approach for generating the cricket news. In cases of languages other than English, the template-driven approach to narrative generation would be particularly useful since there are not enough programming libraries and packages to build the system. In this system, there are two levels of templates defined i.e. sentence template and phrase template. It also discusses the suitability of using an architecture based on typical NLG pipeline approach.

It was noted that the evaluation of an NLG system is challenging due to lack of standardized methods. Therefore, suitable evaluation plan which is performed. According to the evaluation plan there are three scores which are to be calculated to compare actual output of the system with the reference text taken from two websites. They are similarity score, degree of closeness and data count. According evaluation results, the similarity score gives a favourable result when the generate summary was compared with the average written summary. Therefore, it can be concluded that use of templates has a major contribution for getting low percentages for similarity score. Similarly, satisfactory values could be gained for data count and degree of similarity when compared with average written summary. However, evaluating generated summary with expert summary was not successful since with a smaller number of matches, it was not possible to understand a pattern. Therefore, it is better to have human based evaluation to compare the generated summary with an expert written summary.

FUTURE WORK

There are some future research directions in respect to this automated cricket news generating system. Having said that the system is capable of extending it to different languages because of the usage of templates, a Sinhala news generation is also practically possible.

Referring expression generation is one of the NLG tasks where the system applies rules on the input data to determine appropriate places where phrases such as he, she, they, it can be used to replace the corresponding proper nouns.

Content selection is done based on rule-based approach and it also can be done with a learning method if a sizable input is given. In this system cricket commentary is not in use for the cricket news generation and in future it is possible to make use of cricket commentary for input generate in detailed cricket news. News heading can be generated in future. At the moment it was done and since it must contain more creativity. For this a learning method can be followed.

REFERENCES

[1] K. Staykova, "Natural Language Generation and Semantic Technologies", Cybernetics and Information Technologies, vol. 14, no. 2, pp. 3-23, 2014.

[2] Graefe, A. (2016). Guide to automated journalism.

[3] "ESPNcricinfo - Cricket Teams, Scores, Stats, News, Fixtures, Results, Tables", ESPNcricinfo, 2018. [Online]. Available: http://www.cricinfo.com/. [Accessed: 20- Jul- 2018].

[4] E. Reiter and R. Dale, "Building applied natural language generation systems", Natural Language Engineering, vol. 3, no. 1, pp. 57-87, 1997.

[5] A. Gatt and E. Krahmer, "Survey of the State of the Art in Natural Language Generation: Core tasks, applications and evaluation", Journal of Artificial Intelligence Research, vol. 61, pp. 65-170, 2018.

[6] K. Deemter, M. Theune and E. Krahmer, "Real versus Template-Based Natural Language Generation: A False Opposition?", Computational Linguistics, vol. 31, no. 1, pp. 15-24, 2005.

[7] S. Busemann and H. Horacek, "A Flexible Shallow Approach to Text Generation", 9th International Workshop on Natural Language Generation, Niagara-on-the-Lake, Canada, pp. 238-247, 1998.

[8] M. Cristiá and B. Plüss, "Generating Natural Language Descriptions of Z Test Cases", INLG '10 Proceedings of the 6th International Natural Language Generation Conference, pp. 173-177, 2010.

[9] E. Reiter and R. Dale, "Building applied natural language generation systems", Natural Language Engineering, vol. 3, no. 1, pp. 57-87, 1997.

[10] H. Stenzhorn, "XtraGen — A Natural Language Generation System Using XML- and Java-Technologies", 2002.

[11] M. White and T. Caldwell, "Exemplars: Practical, Extensible Framework for Dynamic Text Generation", In Proceedings of the Ninth International Workshop on Natural Language Generation, 1998.

[12] W. Oremus, "The First News Report on the L.A. Earthquake Was Written by a Robot", https://goo.gl/jeHRPX., 2014.

[13] L. Leppänen, M. Munezero, M. Granroth-Wilding and H. Toivonen, "Data-Driven News Generation for Automated Journalism", Proceedings of the 10th International Conference on Natural Language Generation, pp. 188–197, 2018.

[14] M. Theune, E. Klabbers and J. Odijk, "From Data to Speech: A General Approach", Natural Language Engineering, vol. 7, pp. 47-86, 2000.

[15] E. Goldberg, N. Driedger, and R. I. Kittredge. Using natural-language processing to produce weather forecasts. IEEE Expert: Intelligent Systems and Their Applications, 9(2):45–53, April 1994.

[16] J. Aires, "Automatic Generation of Sports News", Postgraduate, University of Porto, 2016.

[17] F. M. Hasan, "Automatic generation of multilingual sports summaries", Postgraduate, Applied Science: School of Computing Science, 2011.

[18] J. Jai Hari Raju, P. Indhu Reka, K. K Nandavi and M. Karky, "Tamil Summary Generation for a Cricket Match", Tamil Computing Lab, College of Engineering Guindy, Anna University, 2011.

[19] K. Bontcheva, "Generating Tailored Textual Summaries from Ontologies", In 2nd European Semantic Web Conference, ESWC 2005, vol. 3532, pp. 241–256, 2019.

[20] E. Reiter, "How to do an NLG Evaluation: Metrics", Ehud Reiter's Blog, 2018.

[21] A. Belz and E. Reiter, Comparing Automatic and Human Evaluation of NLG Systems, 1st ed. Trento, Italy: 11st Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, 2006, pp. 3-7