International Journal of Electrical and Computer Engineering (IJECE)
Vol. 9, No. 6, December 2019, pp. 4929~4938
ISSN: 2088-8708, DOI: 10.11591/ijece.v9i6.pp4929-4938

Automatic summarization of Malayalam documents using clause identification method

Sunitha C[1], A Jaya[2], Amal Ganesh[3]
[1,2]B.S.Abdur Rahman Crescent Institute of Science and Technology, India
[3]Vidya Academy of Science and Technology, India

Article Info

Article history:

Received Jan 12, 2019
Revised Jun 4, 2019
Accepted Jun 26, 2019

Keywords:

Abstractive summarization
Clause boundary
MBT tagger
Morphological analysis
Sentence score

ABSTRACT

Text summarization is an active research area in the field of natural language processing. Huge amount of information in the internet necessitates the development of automatic summarization systems. There are two types of summarization techniques: Extractive and Abstractive. Extractive summarization selects important sentences from the text and produces summary as it is present in the original document. Abstractive summarization systems will provide a summary of the input text as is generated by human beings. Abstractive summary requires semantic analysis of text. Limited works have been carried out in the area of abstractive summarization in Indian languages especially in Malayalam. Only extractive summarization methods are proposed in Malayalam. In this paper, an abstractive summarization system for Malayalam documents using clause identification method is proposed. As part of this research work, a POS tagger and a morphological analyzer for Malayalam words in cricket domain are also developed. The clauses from input sentences are identified using a modified clause identification algorithm. The clauses are then semantically analyzed using an algorithm to identify semantic triples - subject, object and predicate. The score of each clause is then calculated by using feature extraction and the important clauses which are to be included in the summary are selected based on this score. Finally an algorithm is used to generate the sentences from the semantic triples of the selected clauses which is the abstractive summary of input documents.

Corresponding Author:

Sunitha C,
Department of Computer Science and Engineering,
B.S.Abdur Rahman Crescent Institute of Science and Technology,
GST Road, Vandalur, Chennai 600 048, Tamilnadu, India.
Email: sunitha@vidyaacademy.ac.in

1. INTRODUCTION

With the exponential growth of information in the internet, it is very essential to consolidate the related information and to present the contents in a concise manner. In this context, automatic summarization of documents becomes an essential and important activity. Summarization is an ongoing research work in the area of natural language processing. Summarization can be classified into various categories, Extractive summarization and Abstractive Summarization, Single document and Multi document summarization, Generic and Query based summarization etc.

In extractive summarization, the sentences are scored based on some statistical measures such as sentence position, proper noun feature, numeric feature, TF-IDF feature etc. The top scored sentences are then selected to generate summary. The advantage of this method is that the summary includes the original sentences from input text and we are not redefining the sentences. Majority of the works have been carried out in this area. But this method sometimes lack semantical content of the document. In abstractive

summarization [1] the meaning of the sentences are conceptualized and based on this the summary generated. But the difficulty of this method is the lack of efficient techniques to represent the sentences semantically.

In multi document summarization the important sentences related to a particular area/topic from multiple sources are extracted to produce a summary whereas in single document summarization the important sentences/concepts from one document is considered. In generic summarization, the total concept or idea of the document is extracted whereas in query based summarization, the sentences related to the terms in query are selected to produce summary.

A large number of research works in the area of extractive summarization have been carried out in foreign languages, but very few research works happened in the area of abstractive summarization. Due to the agglutinative nature of Dravidian languages, it is very difficult to generate an abstractive summary [2]. Malayalam is one of the Indian languages mainly spoken in Kerala. An effective summarizer is not available in Malayalam due to various reasons. Malayalam language processing is very difficult because of its agglutinative nature and many words are found as compound words. The morphology of the language is highly inflectional, derivative and compounding. There is no upper or lower case for Malayalam letters like English which if present will help to identify pronouns. Also the same word can appear with inflectional and morphological variations in sentences and same concept may be expressed using synonyms in different sentences. Unavailability of freely and publicly available corpora is a major problem in this language. Lack of complete and efficient preprocessing tools in Malayalam makes further research very difficult. Very few research works happened in the area of extractive summarization. An efficient abstractive summarization system for Malayalam is not developed yet.

In this paper, an abstractive summarization system for Malayalam documents using clause identification method is proposed. As there is no efficient abstractive summarizer in Malayalam, this work can be considered as a base towards the research in this area. In this method, after preprocessing the input documents, clauses are identified from the input documents using a modified clause identification algorithm. The clauses are then semantically analyzed using an SOP identification algorithm to extract the semantic triples from the clauses- subject, object and predicate. The score of each clause is then calculated by using feature extraction and the important clauses which are required to include in final summary are selected based on this score. A sentence generation algorithm is used to generate the sentences from the semantic triples of the selected clauses and this will be the final summary. The work is carried out using cricket as the domain. The paper is organized into different sections. Section 2 describes the related works in the area of summarization in Indian languages. Section 3 describes the overall architecture of the system. Section 4 explains the results and discussions. Section 5 is the conclusion.

2. RELATED WORKS

Few research works have been carried out in Indian languages in the area of abstractive summarization. These works can be classified into two: syntactic and semantic approaches. In syntactic summarization, a syntactic parser is used to analyze the text and it lacks the semantic representation of input document. But in semantic approach, the input text is represented semantically.

J. Balaji et al. [3] proposed a semi-supervised bootstrapping approach for the identification of important components for abstractive summarization. In the proposed approach a fully connected semantic graph of a document is given as the input. Here, first semantic graphs are constructed for sentences, which are then connected by synonym concepts and co-referring entities to form a complete semantic graph. The direction of the traversal of nodes is determined by a modified spreading activation algorithm, where the importance of the nodes and edges are decided, based on the node and its connected edges under consideration. From this the most important nodes and edges are selected to form a summary.

Atif Khan et al. [4] proposed a semantic graph based approach with improved ranking algorithm for abstractive summarization of multi-documents. The semantic graph is built from the source documents in such a manner that the graph nodes denote the predicate argument structures (PASs) which are the semantic structure of sentences and are automatically identified by using semantic role labeling. The graph edges represent similarity weight, which is computed from PASs semantic similarity. From this structure, a graph ranking algorithm is used to select the important nodes and edges which can be used to represent the summary.

Atif Khan, Naomie Salim and Yogan Jaya Kumar [5] proposed a framework for abstractive summarization of multi-documents; the method selects contents of summary not from the source document sentences but from the semantic representation of the source documents. In this framework, contents of the source documents are represented by predicate argument structures by employing semantic role labeling. Content selection for summary is made by ranking the predicate argument structures based on optimized features, and using language generation for generating sentences from predicate argument structures.

Rajina Kabeer and Sumam Mary Idicula [6] used both statistical method and semantic graph based method for summarizing Malayalam documents. In statistical sentence scoring method, the important sentences are extracted based on some statistical measures. In semantic graph based method, sentences are converted into clauses. From these clauses subject, object and verbs are extracted. Using these triples, a semantic graph is generated for the whole document. From this graph, a sub graph is generated using semantic graph reduction approach. This subgraph represents the summary sentences to be generated. From the subgraph, the final summary sentences are generated.

Ibrahim F. Moawad et al. [7] presented a novel approach to create an abstractive summary for a single document using a rich semantic graph reducing technique. The approach summaries the input document by creating a rich semantic graph for the original document, reducing the generated graph, and then generating the abstractive summary from the reduced graph.

Muhidin Mohamed, Mourad Oussalah [8] proposed an innovative graph-based text summarization model for generic single and multi-document summarization. The approach involves four unique processing stages: parsing sentences semantically using Semantic Role Labeling (SRL), grouping semantic arguments while matching semantic roles to Wikipedia concepts, constructing a weighted semantic graph for each document and linking its sentences (nodes) through the semantic relatedness of the Wikipedia concepts. An iterative ranking algorithm is then applied to the document graphs to extract the most important sentences deemed as the summary.

Manju K et al [9] proposed graph based multidocument extractive summarization method for Malayalam language similar to LexPageRank. The proposed model uses a weighted undirected graph to represent the documents. The significant sentences for the summary are selected by applying the Page Rank algorithm. Kanitha and Shanavas [10] used statistical graph theoretic approach for Malayalam Text summarization. The sentences are represented as nodes and the relation is represented as edges. The cardinality of a graph shows the importance of sentences. The important summary sentences are selected based on this cardinality by setting a threshold value.

Kavya Kishore et al. [11] in their paper used a suitable semantic representation called Karaka tree for representing the sentences in the document. Karaka tree that is based on Panini's grammar framework is a suitable representation for representing Malayalam sentences as it has resemblance to the Malayalam grammar specification. The Karaka trees constructed are merged based on sentence aggregation rules. Also a sentence extractor module has been used that helps to identify the core ideas in the document using statistical approaches. Therefore the system incorporates the benefits of both extractive and abstractive methods.

Kannada text summarization works by Kallimani et al. [12] mainly deal with statistical approaches. Jayashree et al. proposed Kannada text Summarizer based on key word extraction. Inverse-Document-Frequency techniques with Term-Frequency were applied for extracting the keywords for making summary. Banu M et al [13] used semantic graph reduction approach in their work. Semantic triples Subject, Object and Predicate are extracted from individual sentences to form a semantic graph for the entire document. These semantic triples undergo semantic normalization process to reduce the number of nodes thereby generating a sub graph. This sub graph seves as the basis for generating abstractive summary.

Nikita Munot and Sharvari S. Govilkar [14] proposed a conceptual framework for abstractive text summarization. An approach is presented to generate an abstractive summary for the input document using a graph reduction technique. This paper proposes a system that accepts a document as input and processes the input by building a rich semantic graph and then reducing this graph for generating summary. Sunitha.C et al. [15] tried to identify semantic roles from the text using paninian grammar based on karaka theory. From these semantic roles subject, object and predicate can be identified which will be used for text summarization.

M. John Basha and K.P. Kaliyamurthie [16] proposed an efficient text based clustering framework. After the dataset is preprocessed, the similarities between the words are computed using the cosine similarity. The similarities between the components are compared and the vector data is created. From the vector data the clustering particle is computed. P.V. Amoli [17] proposed method is a summarization-based hybrid algorithm. They preprocessed the text to remove the unimportant words and calculated TF-IDF score of words. After this calculation, clustering is done to form different clusters and from each cluster the more important weight sentences are slelected for summarization.

3. RESEARCH METHOD

The proposed method generates an abstractive summary of Malayalam documents using clause identification method. The input text undergoes some preprocessing steps such as sentence splitting and tokenization. The stem words are generated from these valid tokens using a morphological analyzer. The stem words are checked with a manually developed wordnet to obtain similar concept words if any.

Next a clause identification algorithm is used to find out the clauses. In Malayalam, sentences may contain more than one clause which contains important meaning. Semantic triples are extracted from these clauses. The clauses are then ranked by feature extraction. From these top ranked clauses, the summary sentences are generated by sentence generation method. Limited research works have been carried out in Indian languages in the area of abstractive summarization. These works can be classified into two: syntactic and semantic approaches. In syntactic summarization, a syntactic parser is used to analyze the text and it lacks the semantic representation of input document. Most works are based on syntactic summary. But in semantic approach, the input text is represented semantically. Semantic triples can be used for representing the sentences semantically. Malayalam documents related to cricket domain are collected in the form of a text file. This text file is preprocessed which contains the following steps.

3.1. Overall architecture

The overall architecture of the proposed system is given in Figure 1. Various phases of the proposed system are explained in the following sections.

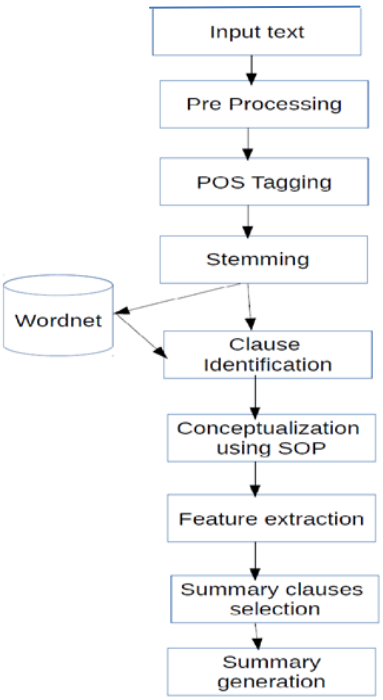

Figure 1. Architecture of proposed abstractive summarization system for Malayalam

3.1.1. Pre processing

Preprocessing is an important activity in any of the natural language applications. Here the document collected is splitted into sentences because sentence level processing is carried out in our method. The sentences are then splitted into tokens. Also filtration is done to remove the special characters. The output of this phase is a sequence of valid tokens.

Consider the following example, സച്ചിൻ ബാറ്റു ചെയ്തു (Sachin battu cheythu.) / (*Sachin did batting*). Here tokens are സച്ചിൻ, ബാറ്റു, ചെയ്തു (Sachin, battu, cheythu). Tokenization is performed by stripping the text using space and delimiters. Based on this space, the sentences are splitted into individual tokens. This phase is implemented using a python program.

3.1.2. POS tagging

The tokens obtained in the previous step are passed through a POS tagger to get an appropriate tag for each token. Part of Speech tagging is the process of assigning a valid tag to each word or token based on trained data set and also based on neighboring words. Even though large number of POS taggers is available for foreign languages, a complete POS tagger for Malayalam language is not available. So we have

developed a POS tagger for Malayalam words in cricket domain. The data set is collected manually and the tagger is trained using a classifier. The implementation of tagger is done using Memory Based Tagger (MBT). The tagset used is based on the BIS tagset.

MBT works on the principle of Memory based learning [18]. It differs from other classifiers in such a way that it learns from experiences instead of extracting rules or makes abstract representations. So it can tag the words based on the surrounding words in the sentence. And also this method uses the concept of reusing memory for remembering that experiences directly. The system is trained with around 10000 tokens in cricket domain.

The Traning format is: സച്ചിൻ (Sachin)- N-NNP
ബാറ്റ് (Bat) – N-NN

3.1.3. Stemming

Stemming is a crucial component in most of the NLP applications. Since the stemming identifies the same stem for all inflectional variants of a lexeme, it will improve the performance of information retrieval systems. In stemming, each token in the sentence having a valid POS tag is converted into its root form. A full fledged stemmer is not available in Malayalam language. To deal with all possible inflections of an agglutinative language like Malayalam, a system is yet to be designed. So we have developed a morphological analyzer to suit for our system.

There are different methodologies such as rule based approach, suffix stripping method, paradigm based approach etc. for generating a morphological analyser..The structure of a word is very important in morphological analyzer. Almost all languages have got some specific structures. Generally any word is a combination of base and suffix. Word=stem+affixes; Stem: morpheme that forms the cenral meaning unit and Affixes: prefix, suffix, circumfix etc.

A suffix stripping based morphological analyzer is developed as part of this work. This suffix stripping based Morphological analyzer for Malayalam deals with all possible inflections of nouns and verbs in Malayalam. Since Malayalam is a language with high rate of inflections and ambiguities, it is not effective to depend only on a dictionary based approach. So a combined rule-cum-dictionary based method is used along with the suffix stripping approach. Stemming is used in our system for extracting similar concept words using a wordnet in cricket domain. In our system stem words are generated for nouns and verbs only, based on their POS tags.

Eg: Root word of സെഞ്ചുറിയുടെ is സെഞ്ചുറി
Root word of അടിച്ചു is അടിക്കുക

3.1.4. Wordnet

In any language, different words which are similar in concepts may appear in sentences frequently. We need to identify those words because they refer to more or less same concept. When we summarize the document, these same meaning sentences/words must be reduced. Also these words can be used to identify similar concept sentences or clauses. For this we have developed a wordnet in Malayalam which contains meanings and synsets of nouns and verbs pertaining to the field of cricket. The root words obtained after stemming is mapped with wordnet to retrieve the synsets. The words with similar synsets are replaced with their common concept so that all these words mapped to the same concept.

Eg: Synonyms of the word മത്സരം are പോരാട്ടം, മാച്ച്
Synonyms of the word ഇന്ത്യ are ഭാരതം, ഹിന്ദുസ്ഥാന്

3.1.5. Conceptualization using SOP

Malayalam sentences may contain more than one clause. Clauses represent a meaningful information part of a sentence. We can semantically process a sentence by extracting the clauses from sentences. Subjects (S), objects (O) and Predicates (P) of a clause are the important semantic components of a clause. From the sentences, clauses can be identified by applying the following rules:

Modified clause identification algorithm

a. Check the POS tags of tokens in the sentences from left to right. If it is a verb with the tag V_VM_VF or an adjectival participle with the tag ADJP, then it is the boundary of a clause.
b. In the case of adjacent verbs (V_VM_VNF or V_VM_VF or V_AUX or ADJP) the last verb is considered for marking boundary.
c. If the POS tag is ADJP, the noun following it (in case of compound nouns or compound proper nouns the group should be considered) along with PSP qualifier is also included in the clause. Also the same noun is to be added to the beginning of next clause in the same sentence if any.

From the clauses generated, Subjects, Objects and Predicates can be identified by applying the following algorithm:
SOP identification algorithm:

a. Predicate: The verb or adjectival participle (identified by V_VM_VF or ADJP) in the clause along with the verb qualifiers will form the predicate. If there are adjacent verb POS tags such as V_VM_VNF or V_VM_VF or V_AUX or ADJP along with the above, they also form part of predicate.
b. Object: If the predicate is a verb (V_VM_VF), the noun preceding it will be the object. If the predicate is an adjectival participle (ADJP), the noun following it will be the object. Rules regarding compound nouns and qualifiers mentioned in the clause identification algorithm are applicable to here also.
c. Subject: If the predicate is a verb, the noun preceding the object (which is not a qualifier of the object) will be the subject. If the predicate is an adjectival participle, the noun preceding the adjectival participle will be the subject. If there is no noun preceding the object in the same clause,
    - if the predicate of the preceding clause is an adjectival Participle, its object will form the subject else subject of the preceding clause will be the subject of the current clause.

3.1.6. Feature extraction

The most important task in summarization is to select the important sentences from the input document which form a summary. The importance of sentences is calculated by analysing the importance of the clauses generated from the sentences. This is done by calculating some statistical features of clauses generated from sentences. The features are extracted from the clauses and the weighted average of all these features are used for calculating the clause score. The features like clause position, number of numeric data, no. of proper nouns, TF-IDF frequency and no. of title words are used in our implementation which is explained below.

3.1.6.1. Clause position feature

Clause Position is the position of a sentence which includes that clause in a document. This feature is used because in most cases the important sentences lie in the first and last portion of the document. So these sentences and thereby clauses also have more chances to include in the summary. The value of this feature is normalized to a scale of 0 and 1. It is calculated as per the equation, PositionF= (maxpos −curpos+1) /maxpos, where maxpos is the maximum number of clauses in the document and curpos is the position of the clause in the document.

3.1.6.2. Numeric value feature

The sentences containing numerical data are relevant as it indicates event related attributes like time of occurrence, population, statistical data, etc., and are most probably to be included in the summary. The score is calculated as the ratio of number of numerical data in the clause to length of clause.

$$\text{NumF} = \frac{\text{No. of numerical data} \in C_i}{\text{Length of clause } C_i}$$

3.1.6.3. Proper noun feature

As the proper nouns indicate the name of person or place etc the clauses which contain the proper nouns are more important than others. This feature is calculated based on the POS tag of tokens in the sentences. The score of a clause i, C_i is calculated as the ratio of number of proper nouns in the clause to the length of the clause.

$$\text{Proper NounF} = \frac{\text{No. of propernouns} \in Ci}{\text{Length of sentence Ci}}$$

3.1.6.4. TF-IDF feature

The goodness of a sentence is usually represented by the importance of the words present in it. TF-IDF is a simple but powerful heuristic for ranking the sentences according to their importance. A Vector Space model is built at the sentence level by grouping all the sentences of the documents. Now for scoring the clauses, we determine the TF-IDF of each clause in a document. TF-IDF is calculated usimg the following rules.

a. Calculate TF of a term which is defined as the no. of occurrences of the term in the clause / total no. of words in the clause.
b. Calculate IDF of a term which is defined as ln(N/Nt) where N is the total number of clauses in the document and Nt is the no. of clauses which contain the term t.
c. Calculate TF-IDF of each term in the clause as TF * IDF
d. Take the sum of TF-IDF of all terms in the clause and this is the TF-IDF score of that clause.

3.1.6.5. Title word feature

The presence of title words in a clause makes the clause more important. The feature is calculated as follows.

$$\text{TitleF} = \frac{\text{No. of title words} \in Ci}{\text{Total no. of title words}}$$

3.1.7. Summary clauses selection

After obtaining the score of all features explained above pertaining to clauses, a weighted average of the score is calculated using the Table 1.

Table 1. Weight of features

Feature	Weight
TF-IDF Score	10
Title words	8
Proper Nouns	7
First Paragraph	6
Last Paragraph	5
Numeric Value	4

The overall score of a clause C based on the features will be,

$$\text{Score}(C) = \sum_{i=1}^{n} w_i * F_i$$

Now we have a key, value pair consisting of clauses and its corresponding scores. Sort the clauses based on clause score. From these set of clauses select the clauses which are to be included in summary. The selection can be done either based on the score or count. In this work we have selected half of the total number of clauses for inclusion in summary as the resulting summary is more meaningful in this case.

3.1.8. Summary generation

Subjects, Objects and Predicates generated from the clauses are restructured into sentences by applying the following rules. If all the clauses of the sentence is present, the same sentence can be reproduced else the following rule is used for generating sentence.

a. If the verb is ADJP, the clause is converted into a sentence in the order subject, root form of object and past tense of verb along with qualifiers.
b. If the verb is V_VM_VF or V_VM_VNF, then the clause is converted into a sentence in the order subject, object and past tense of the verb.

4. RESULTS AND DISCUSSIONS

We have tested our system with 20 sets of Malayalam on line news documents collected from Malayala Manorama on line newspaper. The summary is generated for each of the documents using our system. The summary is also generated manually. The results are promising and the summary is almost similar to human generated summary. As there is no effective abstractive summarization system in Malayalam, this work can be considered as the first step towards abstractive summarization in Malayalam. The system is implemented with cricket as the domain. The summary can be further improved by enriching the training data set for POS tags and Morphological analyzer. Also the approach can be extended to all types of documents with a full fledged POS tagset and morphological analyzer. The use of Wordnet in our system helped to identify similar meaning sentences which will improve TF-IDF score and thereby increasing

chances of inclusion in final summary. So the similarity can be increased by incorporating more words in wordnet also the clauses which contain proper nouns and numerical figures will have more chances to be included in summary. With the clauses we can extract the overall semantic content from the document and using this we can represent abstractive summary. A sample input and output is given below:

Input text

ക്രിക്കറ്റിൽ ഇംഗ്ലണ്ടിനെ അട്ടിമറിച്ച് സ്കോട്ടിഷ് പടയോട്ടം.

ഏകദിന ക്രിക്കറ്റിന്റെ ചരിത്രത്തിലെ ഏറ്റവും വലിയ അട്ടിമറികളിലൊന്നിൽ ലോക ഒന്നാംനമ്പർ ടീമായ ഇംഗ്ലണ്ടിനെ കുഞ്ഞന്മാരായ സ്കോട്ലൻഡ് വീഴ്ത്തി. ആവേശം അടിമുടി നിറഞ്ഞ പോരാട്ടത്തിൽ ആറു റൺസിനായിരുന്നു സ്കോട് ലൻഡിന്റെ ക്രിക്കറ്റ് ഭാവിക്ക് ഊർജമേകിയ വിജയം. ആദ്യം ബാറ്റു ചെയ്ത സ്കോട് ലൻഡ് അഞ്ചു വിക്കറ്റിന് 371 റൺസെടുത്തപ്പോൾ ഇംഗ്ലണ്ടിന്റെ ഇന്നിങ്സ് 365 റൺസിൽ അവസാനിച്ചു. കാലും മക്‌ലിയോഡ് പുറത്താകാതെ നേടിയ 140 റൺസാണ് സ്കോട്ടിഷ് ഇന്നിങ്സിന്റെ അടിത്തറ. 105 റൺസെടുത്ത ബെയർസ്റ്റോയിലൂടെ തിരിച്ചടിക്കാനുള്ള ഇംഗ്ലണ്ടിന്റെ ശ്രമം മധ്യനിരയുടെ തകർച്ചമൂലം യാഥാർഥ്യമായില്ല.ക്യാപ്റ്റൻ കൈൽ കോട്സറും(58) മാത്യു ക്രോസും(48) ചേർന്ന് സ്കോട്ലൻഡിന് സെഞ്ചുറി കൂട്ടുകെട്ടോടെ ഉജ്വല തുടക്കം സമ്മാനിച്ചു. എന്നാൽ രണ്ടുപേരെയും തുടർച്ചയായി നഷ്ടമാകുമ്പോൾ അവർക്കു 107 റൺസ്. മക്‌ലിയോഡും ജോർജ് മുൻസെയും(55) ചേർന്നു നാലാം വിക്കറ്റിൽ 107 റൺസെടുത്തതോടെ സ്കോട്ലൻഡ് വീണ്ടും കുതിപ്പിന്റെ വഴിയിൽ. ഡർഹം ബാറ്റ്സ്മാനായ മക്‌ലിയോഡ് 70 പന്തുകളിൽ 100 റൺസ് കടന്നു. മൊത്തം 94 പന്തുകൾ നേരിട്ട മക്‌ലിയോഡ് 16 ബൗണ്ടറിയും മൂന്നു സിക്സറുമടക്കമാണ് 140 റൺസിലെത്തിയത്. ഇംഗ്ലണ്ട് നിരയിൽ ഭേദപ്പെട്ട പ്രകടനം നടത്തിയ ബോളർ മോയിൻ അലി പോലും 10 ഓവറിൽ 66 റൺസ് വഴങ്ങി. പ്ലങ്കറ്റിന്റെ പത്തോവറിൽ നിന്ന് സ്കോട്ലൻഡ് 85 റൺസ് സ്വന്തമാക്കി.ഒന്നാം വിക്കറ്റിൽ 129 റൺസെടുത്ത ഇംഗ്ലണ്ടും നല്ല തുടക്കം കുറിച്ചു. 34 റൺസെടുത്ത ജാസൻ റോയ് ആണ് ആദ്യം പുറത്തായത്. പിന്നീട് ബെയർ സ്റ്റോയ്ക്കൊപ്പം അർധ സെഞ്ചുറിയോടെ അലക്സ് ഹെയ്ൽസ്(52) പട നയിച്ചു. പിന്നീടെത്തിയവർ മികച്ച തുടക്കം മുതലാക്കാൻ കഴിയാതെ പുറത്തായതോടെ ഇംഗ്ലണ്ട് സ്കോർ ഏഴിന് 276 റൺസിലെത്തി. എട്ടാം വിക്കറ്റിൽ മോയിൻ അലിയും(46) പ്ലങ്കറ്റും(47) ചേർന്ന് 71 റൺസോടെ വീണ്ടും പ്രതീക്ഷ നൽകിയെങ്കിലും 347 റൺസിൽ അലി പുറത്തായതോടെ സ്കോട്ലൻഡിനായി മേൽക്കൈ. 48.5 ഓവറിൽ ഇംഗ്ലണ്ടിന്റെ ഇന്നിങ്സ് അവസാനിച്ചു.ഏകദിനത്തിൽ രാജ്യാന്തര ക്രിക്കറ്റ് കൗൺസിലിന്റെ ഒരു അസോഷ്യേറ്റ് അംഗം കുറിക്കുന്ന ഏറ്റവും വലിയ സ്കോറാണു സ്കോട്ലൻഡിന്റെ 371 റൺസ്. 1997ൽ ബംഗ്ലദേശിനെതിരെ മൂന്നു വിക്കറ്റ് നഷ്ടത്തിൽ കെനിയ കുറിച്ച 347 റൺസ് ആയിരുന്നു ഇതുവരെ റെക്കോർഡ്. 2014ൽ കാനഡയ്ക്കെതിരെ നേടിയ 341 റൺസ് ആയിരുന്നു സ്കോട് ലൻഡിന്റെ ഇതുവരെയുള്ള മികച്ച സ്കോർ.

Abstractive summary

ഏകദിന ക്രിക്കറ്റിന്റെ ചരിത്രത്തിലെ ഏറ്റവും വലിയ അട്ടിമറികളിലൊന്നിൽ ലോക ഒന്നാംനമ്പർ ടീമായ ഇംഗ്ലണ്ടിനെ കുഞ്ഞന്മാരായ സ്കോട്ലൻഡ് വീഴ്ത്തി. കാലും മക്‌ലിയോഡ് പുറത്താകാതെ 140 റൺസ് നേടി. ക്യാപ്റ്റൻ കൈൽ കോട്സറും(58) മാത്യു ക്രോസും(48) ചേർന്ന് സ്കോട്ലൻഡിന് സെഞ്ചുറി കൂട്ടുകെട്ടോടെ ഉജ്വല തുടക്കം സമ്മാനിച്ചു. ഡർഹം ബാറ്റ്സ്മാനായ മക്‌ലിയോഡ് 70 പന്തുകളിൽ 100 റൺസ് കടന്നു. മൊത്തം 94 പന്തുകൾ മക്‌ലിയോഡ് നേരിട്ടു. 16 ബൗണ്ടറിയും മൂന്നു സിക്സറുമടക്കമാണ് 140 റൺസിലെത്തിയത്. ഇംഗ്ലണ്ട് നിരയിൽ ഭേദപ്പെട്ട പ്രകടനം

ബോളർ മോയിൻ അലി നടത്തി. 10 ഓവറിൽ 66 റൺസ് വഴങ്ങി. പിന്നീട് ബെയർ സ്റ്റോയ്ക്കൊപ്പം അർധ സെഞ്ചുറിയോടെ അലക്സ് ഹെയ്ൽസ്(52) പട നയിച്ചു . എട്ടാം വിക്കറ്റിൽ മോയിൻ അലിയും(46) പ്ലങ്കറ്റും(47) ചേർന്ന് 71 റൺസോടെ വീണ്ടും പ്രതീക്ഷ നൽകി. 347 റൺസിൽ അലി പുറത്തായി. ഏകദിനത്തിൽ രാജ്യാന്തര ക്രിക്കറ്റ് കൗൺസിലിന്റെ ഒരു അസോഷ്യേറ്റ് അംഗം കുറിക്കുന്ന ഏറ്റവും വലിയ സ്കോറാണു സ്കോട്ലൻഡിന്റെ 371 റൺസ് . 1997ൽ ബംഗ്ലദേശിനെതിരെ മൂന്നു വിക്കറ്റ് നഷ്ടത്തിൽ കെനിയ 347 റൺസ് കുറിച്ചു.

5. CONCLUSION

Automatic Summarization of documents is very useful in the context of the presence of huge volume of data. Limited works have been carried out in Indian languages due to its agglutinative nature and non availability of standard preprocessing tools. Most research works are based on extractive summarization. But abstractive summarization is closer to the human generated summary, but it requires semantic analysis of the document in which very few research works have been reported. In this paper we tried to implement an abstractive summarization system for Malayalam documents using conceptualization of clauses with cricket as the domain. The clauses are identified from the sentences using a modified clause identification algorithm and the important clauses are then selected using feature extraction and score calculation. The semantic triples – subject, object and predicate- are extracted from clause using rules which can be used to generate the final summary.

REFERENCES

[1] A. Khan and N. Salim, "A Survey on Abstractive Summarization Methods," *Journal of Theoretical and Applied Information Technology*, vol. 59, 2014.

[2] Sunitha C., et al., "A Study on Abstractive Summarization Techniques in Indian Languages," *International open access journal Elsevier Proceedia Computer Science*, vol. 87, 2016.

[3] J. Balaji and T. V. Geetha, "Abstractive Summarization: A Hybrid Approach for the Compression of Semantic Graphs," *International Journal on Semantic Web and Information Systems*, vol. 12, 2016.

[4] A. Khan, et al., "Abstractive Text Summarization based on Improved Semantic Graph Approach," *International Journal of Parallel Programming*, 2018.

[5] A. Khan, et al., "A framework for multi-document abstractive summarization based on semantic role labelling," *Applied Soft Computing*, vol. 30, pp. 737-747, 2015.

[6] R. Kabeer and S. M. Idicula, "Text Summarization for Malayalm Documents - An Experience," *International Conference on Data Science & Engineering (ICDSE)*, 2014.

[7] I. F. Moawad and M. Aref, "Semantic Graph Reduction Approach for Abstractive Text Summarization," *IEEE Seventh International Conference on Computer Engineering & Systems (ICCES)*, 2012.

[8] M. Mohamed and M. Oussalah, "An Iterative Graph-based Generic Single and Multi Document Summarization Approach using Semantic Role Labeling and Wikipedia Concepts," *2016 IEEE Second International Conference on Big Data Computing Service and Applications*, 2016.

[9] Manju K., et al., "Graph based Extractive Multi-document Summarizer for Malayalam-an Experiment," *Proceedings of the World Congress on Engineering 2016, WCE 2016*, vol. 1, 2016.

[10] Kanitha D. K., et al., "Malayalam Text Summarization Using Graph Based Method," *International Journal of Computer Science and Information Technologies*, vol. 9, pp. 40-44, 2018.

[11] K. Kishore, et al., "Document Summarization in Malayalam with sentence framing," *IEEE international conference on Information Science (ICIS)*, 2016.

[12] J. S. Kallimani, et al., "Information Extraction by an Abstractive Text Summarization for an Indian Regional Language," *Natural Language Processing and Knowledge Engineering (NLP-KE), 7th International Conference*, 2011.

[13] Banu M., et al., "Tamil Document Summarization Using Semantic Graph Method," *International Conference on Computational Intelligence and Multimedia Applications*, 2007.

[14] N. Munot and S. S. Govilkar, "Conceptual Framework for Abstractive Summarization," *International Journal on Natural Language Computing (IJNLC)*, vol. 4, 2015.

[15] Sunitha C, et al., "Semantic Role Labeling of Malayalam Web Documents in Cricket domain," *Journal of Theoretical and Applied Information Technology*, vol. 96, 2018.

[16] M. J. Basha and K. P. Kaliyamurthie, "An Improved Similarity Matching based Clustering Framework for Short and Sentence Level Text," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 7, pp. 551-558, 2017.

[17] P. V. Amoli and O. S. Sh, "Scientific Documents Clustering Based on Text Summarization," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 5, pp. 782-787, 2015.

[18] R. Jesuraj and P. C. R. Raj, "MBLP approach applied to pos tagging in Malayalam language," *NCILC*, 2013.

BIOGRAPHIES OF AUTHORS

Sunitha. C pursuing her Ph.D. in Computer Science and Engineering from B.S.Abdur Rahman Crescent Institute of Science & Technology, Chennai and she is also working as an Associate Professor with the department of Computer Science and Engineering at Vidya Academy of Science & Technology, Thrissur, Kerala. Her research interest includes AI, NLP, Text Mining and Big Data Analytics.
Email : sunitha@vidyaacademy.ac.in

Dr. A. Jaya, Professor and Head of Department of Computer Applications at B.S.Abdur Rahman Crescent Institute of Science & Technology, Chennai. Her research interest includes AI, CBR, Ontology, Web mining, Knowledge mining etc. She is guiding many research scholars and published more than 50 international journals.
Email : jayavenkat2007@gmail.com

Amal Ganesh is working as an Assistant Professor, with the department of Computer Science and Engineering at Vidya Academy of Science & Technology, Thrissur, Kerala. His research interest includes AI. NLP, Text Mining and Big Data Analytics.
Email : amal.ganesh@vidyaacademy.ac.in