Sādhanā (2020) 45:32
https://doi.org/10.1007/s12046-019-1248-0

A novel extractive text summarization system with self-organizing map clustering and entity recognition

M RAHUL RAJ[1,*], ROSNA P HAROON[1] and N V SOBHANA[2]

[1]Department of Computer Science and Engineering, Ilahia College of Engineering & Technology, Muvattupuzha, India
[2]Department of Computer Science, Rajiv Gandhi Institute of Technology, Kottayam, India
e-mail: rahulraj2k16.m@gmail.com; rosna.shihab@gmail.com; sobhana.nv@gmail.com

MS received 7 May 2017; revised 3 July 2019; accepted 10 September 2019; published online 25 January 2020

Abstract. Extractive text summarization yields the sensitive parts of the document by neglecting the irrelevant and redundant information. In this paper, we propose a new strategy for extractive single-document summarization in Malayalam. Initially, entity recognition is done, followed by relevance analysis is made based on some context-aware features. The scored sentences are then clustered using self-organizing maps (SOM) and from these clusters, relevant sentences are extracted out based on the proposed algorithm. Both theoretical and practical evaluations are done to analyze the implemented system. In theoretical evaluation, gradient calculations of relevance equations are used to know that which of these sentence scoring features are contributing more. The relevance equation is optimized with the help of Lagrange's multiplier. The complexity analysis of the proposed algorithms is also performed. In practical evaluation, the system compared with online and offline summarizers upon metrics like precision, recall, and F-measure. The system is tested through a non-clustering approach also in order to analyze the impact of clustering used in our work. Some existing strategies like question game evaluation, sentence rank evaluation, and keyword association are also done to evaluate the different parameters like the relevance of sentences, important entity words, etc.

Keywords. Natural language processing; self organizing maps; semantic role labeling; relevance; redundancy; text summarization; entity recognition.

1. Introduction

Text summarization has been always a hot topic in the branch of natural language processing. In this fast world, no one has time to waste by reading long articles. Everybody is interested in getting more information in very less time [1], here comes the importance of summarization. A summary is actually a condensed form of the original document provided that all the information contained in the original document should also present in the summary and which must be small in size [2].

Malayalam is one of the dominating languages in South India with 33 million native speakers and which is the official language of Lakshadweep (a union territory in India). It is also widely spoken language in gulf countries due to the high rate of immigrants from Kerala. Within these few years, a lot of websites and news feeds are being designed in Malayalam. Several texting platforms (like Whatsapp), typing tools (like Google input tool) provide the facility to type in Malayalam. Furthermore, the Kerala state government has made Malayalam as the official language of the government which implies all regulations, acts, circulars and other documentation should be made and communicated in Malayalam [3]. Due to these reasons, the text processing systems are being made rapidly in Malayalam, out of which text summarization is considered with vital importance.

The existing text summarizing systems in Malayalam have several drawbacks such as high computational cost, time and storage issues [4, 5], etc. Furthermore, some of them are based on graph theory [6, 7]. Even though the graph-based method is a powerful solution for several computational tasks, sometimes the complexity of the method reduces performance. In the case of natural language processing, the documents, sentences, phrases or even words are represented as the nodes and edges of a graph. In the case of large documents, the graph created would be larger with more nodes and if it became a complex network traditional algorithms would not give a better performance. But the size of the graph can be only known at the execution time. This will create serious performance issues. The main two problems related to summarization are relevance and redundancy. In the case of extractive summarizer, it is desirable that the most important sentences from the input should be placed in the

*For correspondence

output. How to find this relevance is a problem. Most of the existing systems [8, 9] do this relevance analysis without considering the meaning of the document. It would be efficient if we are able to find the semantic domain of the topic of the document. If there are two or more sentences having the same meaning and they are appearing in the summary, then we can say that the summary contains redundant information. The redundancy can be avoided by clustering methods. In the Malayalam language, there is not enough summarizing system acclaiming the above-stated features. Here comes the importance of a new strategy.

We propose a method of extractive single document text summarizer in Malayalam which is using the concept of semantic role labeling and self-organizing maps (SOM). After preprocessing we apply entity recognition on sentences to mark sentences based on their similarity with the predefined entity words in the dictionary. Then sentence scoring is done by context-aware features. After all, clustering is performed with the help of SOM. What are new in our method are:

1) We introduce SRL in Malayalam text summarization for the first time.
2) We introduce entity recognition in Malayalam text summarization for the first time.
3) All our sentence scoring methods are based on the meaning identification of the document.
4) We are introducing SOM for the first time in extractive single-document summarization.

2. Background

Summarization methods can be divided into several categories. First and foremost is the extractive and abstractive summarization. In extractive summarization, the most relevant sentences of the input go into the output, but in the abstractive summarization new sentences are created from the existing once and at the end simplify the input [8]. The organization of this section is in such a way that we list out the concepts of available papers in Malayalam text summarization, semantic role labeling, and clustering mechanism.

Graph-based methods work good for keeping accuracy. Parsing the sentences, creating semantic graphs and reduction of that graph will give a summary [9]. Using a graph theoretical approach [4] it is easier to partition the document into subtopics by creating subgraphs as well. The relevant sentence identification is also easy by looking into the graph nodes. Maximum marginal reference [10] is the technique whereby the number of sentences in the output can be determined by unit step functions. Such methods are also used in the area of text summarization [11]. Calculation of length score, similarity score, sentence score [9], etc is useful in ranking sentences.

Semantic role labeling [12, 13] is used to categorize the nouns and give specific meaning to the sentences, which are useful in natural language processing (NLP) applications. Construction of a heterogeneous [14] tree with sentences, nouns and semantic frames as nodes followed by ranking nodes based on their relevance and similarity to other nodes will eventually extract the important sentences as the summary result. The creation of semantic frames leads to the creation of a similarity tree and applying the page rank algorithm to rank the sentences is also a very good summarization mechanism. The Karaka theory of Paninian grammar [15, 16] is useful while doing semantic parsing in Malayalam. It gives an idea about the construction of Malayalam sentences from which we can identify semantic roles.

Clustering-based summarization method uses cosine similarity-based clustering [17], Jaccard similarity and features based method [18], sentence similarity-based clustering [19], clustering using BIRCH [20, 21], Support vector machine (SVM) based summarization [22], and clustering using Particle swarm optimization (PSO) [23]. Clustering itself is a process of reordering and grouping the data based on their similarity, which improves the quality of summary by avoiding redundancy.

Self-organizing maps (SOM) [24, 25] are a neural network-based algorithm used for text clustering. It is initially used in the area of image processing [26, 27]. SOM can be visualized as a collection of nodes in a multi-dimensional plane and form clusters. Their movement is based on the distance calculation equation [28]. One of the main advantages of SOM is that it does an unsupervised clustering [29]. Not only in the field of NLP but also it uses in the medical field, classification of financial data and even more areas of science and technology [30].

3. Methods

This section contains an explanation of each step of the entire architecture followed by a working example. The overall architecture of the system is given in figure 1. After input reading phase preprocessing is done. Then we

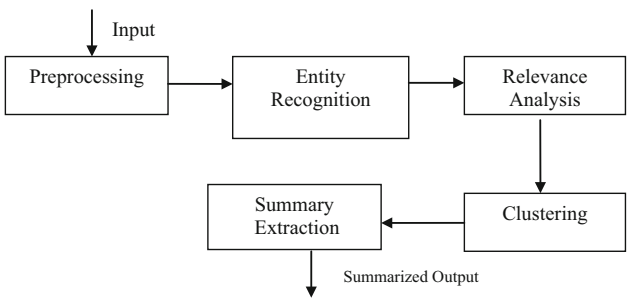

Fig. 1. Overall architecture.

proceed with recognizing entities. Entities are nothing but a collection of words based on topics. Afterward, relevance analysis of individual sentences will be carried out. The next step is the clustering of sentences followed by the summary calculation. Table 1 gives the necessary notations used throughout this document.

3.1 *Preprocessing*

The preprocessing combined of four phases namely sentence extraction, eliminating special character, stemming and stop word removal. Sentence extraction is the process of extracting each and every sentence from the input text because here onwards we are going to deal with the sentences as the fundamental units. Special characters are not at all concerned in summarization process, so remove them from each sentence. Stemming is the process of finding root words from the actual word. For example, consider the word "മരത്തിൽ" it would become "മരം" after stemming. The stemming is necessary because a word may exist in different forms in the same document due to grammatical modifications like "മരത്തിലേക്ക്, മരത്തിൽ, മരത്തിന്റെ, etc". After the process of stemming all these forms will be shrinks to the root word "മരം". For stemming operation the method called LALITHA [31] is being used, root words are selected from Malayalam dictionary 'Olam' [32] and Malayalam word Net [33]. Stop words are the words which have not contributing special meaning to sentences such as "കൂടാതെ, മാത്രം, പക്ഷെ, etc". These words are actually part of grammar and do not contribute to the meaning of the sentence. So it is better to remove them from the document to avoid unnecessary calculation overhead. In short, we can say that stemming and stop word removal are necessary to avoid assignment of false scores to the sentences due to the similarity in grammatical elements of the language and to deal with only root words in the further proceedings.

3.2 *Entity recognition*

Entity recognition [34] is the process in which finding the entity words in the document. An entity is nothing but a collection of words based on some topics. For example, If "politics" is the entity, then the words under this entity are "party", "democracy", etc. Words in the document can be classified into three categories of entities. They are given in table 2.

The system will process in such a way that it will first find to which entity most of the word maps, and set that entity as a primary entity. The words belong to the entity are annotated as ENT i.e. entity words. A few words belong to other entities, label them as SNT i.e. sub-entity words. Yet, there are some words that dos not belong to any entity label them as NTN i.e. non-entity words. The assignment of the entity category is to find some words related to the topics of the document because we are proceeding with the assumption that the words related to the topic convey more information.

3.3 *Relevance analysis*

This is the most crucial step of the system where the importance of sentences is calculated. The main objective of this step is to give a score to each sentence by a number of sentence scoring features. At the end of this step, sentences will have some value attached to it showing the relevance of that sentence in the corresponding document. Those sentence scoring features are explained in the following paragraphs. The notation $\boldsymbol{\rho}$ $(\mathbf{S_i})$ is used to denote the

Table 1. Notations used.

Symbols	Meaning
S_i	A sentence with an identification number. Ex: S_1, S_2etc
Φ_i	A semantic frame after the SRL processing
$S_c^{i,\ j}$	Sentence of c th cluster, ith level with j as priority number
α	Similarity factor taken as 0.1
β	Output factor taken as 0.2
ρ	Relevance factor
q_n	Number of levels in each cluster **n**
far_dist_c	Distance between lowest level and highest level sentences in a cluster
min_dist_c	Fraction of far_dist_c
num_sentence	Number of vectors in the vector space
out_count	Number of sentences in the output
out_sentences $\{\}_{k=1\ to\ r}$	Set of **r** number of sentences selected as summary indexed by **k**
dist (a, b)	Distance between the two sentence vectors **a, b** in vector space
remove(S_c^i)	Remove sentence $\mathbf{S_c^i}$ from the corresponding cluster
order(S_c^i)	Arranging sentences in the descending order of relevance factor

Table 2. Entity types.

Type	Elaboration	Explanation
ENT	Entity words	The system would rank a word as ENT if the word belongs to the most referenced entity. Example: If the document is a chapter in the medical manuscript, then the document belongs to the entity 'medical'.
SNT	Sub-entity words	If there are any other topics the document deals with then the sub-entity is based on those least specified content. Example: If the document is dealing with 'Heart problems' then it's main entity is 'medicine' and sub-entity is 'health'
NTN	Non-entity words	The document contains words that do not belong to any entity specified by the system, and then those are non-entity words. If the document is dealing with 'Heart problems' then its main entity is 'medicine' and sub-entity is 'health' and words like 'explanation', 'define', 'conclude' are belong to non-entity words.

relevance of a particular sentence $\mathbf{S_i}$.ie we can express the relevance as given in equation (1).

$$\boldsymbol{\rho}(\mathbf{S_i}) = \sum_{j=0}^{n} \boldsymbol{feature}_j(\boldsymbol{S_i}) \tag{1}$$

The overall feature of a sentence is the sum of the score assigned by different features. These features are namely sentence entity score, frequent patterns score, and semantic similarity score.

3.3a Sentence entity score

Based on the entity categorization performed in the previous step, we are assigning scores for each category, but before that word ranking is to be carried out, this is done by Eq. (2).

$$\mathbf{rank}(\mathbf{W_i}) = \frac{\|\mathbf{W_i}\|}{\left\|\sum_{j=0}^{n} \mathbf{W_j}\right\|}, \tag{2}$$

where modulus gives the number of occurrences of the word. The numerator contains the number of occurrences of a particular word Wi in the document and the total number of words in the denominator. This ranking is utilized in the sentence entity score. The score for different types of roles are given in table 3. The score of ENT is always greater than SNT and NTN also score of SNT is greater than NTN, because in our context ENT words are more similar to the subject of the document so that importance should be given to these words. We have developed an entity recognizer for this purpose.

Now scoring of sentences takes place. The score of a sentence is the sum of product of rank of word and score of entity type as given in Eq. (3).

$$\text{sentence\_entity\_score}\,(\text{Si}) = \sum_{i=1}^{n} \text{rank}(\text{W}_\text{i}) * \text{score}\left(\text{W}_{\text{i[type]}}\right) \tag{3}$$

Where n is the number of words in the sentence. Using this, score of sentence is calculated.

3.3b Frequent pattern score

The frequent pattern is the most repeating word sequences in the documents. Finding out such sequences and scoring the sentences which contain the frequent pattern are done in frequent_Pattern_score as given in Eq. (4).

$$\text{frequent\_pattern\_score}\,(\text{S}_\text{i}) = \sum_{j=1}^{n} \text{score}\left(\text{Pattern}_\text{j}\right) \tag{4}$$

The first step in this scoring is finding frequent patterns. The question comes is how to identify whether a pattern is a frequent pattern or not. The system does this by applying a mechanism similar to support in association rule mining. Here support is fixed as two, i.e. any pattern which repeats two or more than two considered as a frequent pattern. The next step is the process of determining the score for those patterns. This system applies a length-based scoring strategy for patterns. The scoring strategy of the pattern is given in the table 4. Ten percentage of pattern size is calculated as the pattern score, where pattern size is the number of words in the pattern.

3.3c Semantic similarity score

Semantic similarity is calculated on the basis of semantic role labeling (SRL). SRL is the process of assigning different roles to the nouns in a sentence. Each sentence is divided into one or more semantic frames, in which a

Table 3. Entity type scoring strategy.

Type	Score
ENT: Entity words	0.7
SNT: Sub entity	0.4
NTN: Not Entity	0.0

Table 4. Pattern scoring strategy.

Pattern size	Score
2	0.2
3	0.3
4	0.4

Table 5. Semantic roles.

Role	Elaboration	Explanation
AM-THEME	Entity words	They undergo action, but do not change the state. Ex: We believe in God; Here God is a theme because no operation changes its state.
AM-LOC	Sub-entity words	Denotes current location. Ex: I am in school now. Her school is the location.
AM-TMP	Non-entity words	Denotes time. Ex: We met yesterday. Here yesterday is time.

predicate (verb) and one or more arguments (nouns) are present [12]. Here SRL tuples are generated so as to rank the sentences and cluster them easily. A semantic frame is a collection of words with necessary annotations. Each frame contains a verb followed by a sequence of nouns with role assignment. The role is nothing but a specified meaning already available in the semantic role labeller. In this paper, we have created our own semantic role labeller (SRL) with a proper dictionary and grammatical rules present in the Malayalam language. Consider that from the above step we have extracted N sentences that are represented as $\mathbf{S_1}$, $\mathbf{S_2}$,…$\mathbf{S_N}$. Correspondingly, there is **M** number of frames represented as $\mathbf{\Phi_1}$, $\mathbf{\Phi_2}$,… $\mathbf{\Phi_M}$ created, provided that $\mathbf{N \leq M}$. Sometimes, two or more semantic frames may be generated from a long sentence denoted as Φ1 and Φ2 in figure 5. Normally semantic role labeller has some predefined roles such as AM-TMP denotes time; AM-LOC denotes location and so forth. Here we are using three main roles. They are given in table 5. In Malayalam language, SRL is done by using the Karaka theory in Paninian grammar [16].

$$\text{semantic\_similarity\_score}(F_i) = \sum_{i=1}^{n} \text{simRole}(A_i, A_j) + \text{simArg}(A_i, A_j) \quad (5)$$

The semantic similarity score first applied for semantic frames. The similarity between each and every frame is calculated using Eq. (5). simRole () is the method that finds whether the role labeling of arguments is the same or not and simArg () method finds whether those arguments are the same or not. Note that the frames belong to the same sentence would not be compared at any cost. It is because of the reason that we are comparing sentences not within sentences.

$$\text{semantic\_similarity\_score}(S_i) = \sum_{j=1}^{n} \text{semantic\_similarity\_score}(F_j) \quad (6)$$

After finding the frame similarity score, the sum of such frame similarity is accumulated to find semantic similarity of sentences as given in Eq. (6). The overall score of a sentence will be the sum of sentence_entity_score (), frequent_pattern_score () and semantic_simialarity_score (), which is given in Eq. (7).

$$\rho(S_i) = \text{sentence\_entity\_score}(S_i) + \text{frequent\_pattern\_score}(S_i) + \text{semantic\_similarity\_score}(S_i) \quad (7)$$

3.4 *Clustering*

Self-Organizing Maps [26] is used as a clustering algorithm here. Clustering will partition the sentences into m clusters. Each cluster is having a different number of sentences or we can say that each cluster contains sub-topics of the document. The need for clustering is to avoid redundancy because the sentences in a particular cluster have high similarity to each other. In the process of summarization, some of these sentences will be selected for summary generation provided that more similar contents would not be in the summary. The working of SOM is as described in figure 2. The first step is to create the map nodes as needed. Map nodes are nothing but arbitrary vectors created by the programmer which acts as the cluster centers. The next step is to read all the input vectors. For each map, the node finds an input vector that is close to it and name this as the best matching unit (BMU). Now update the position of BMU, and neighbors of BMU according to Eq. (8), in which s is the current iteration, u is the index of BMU and v is the

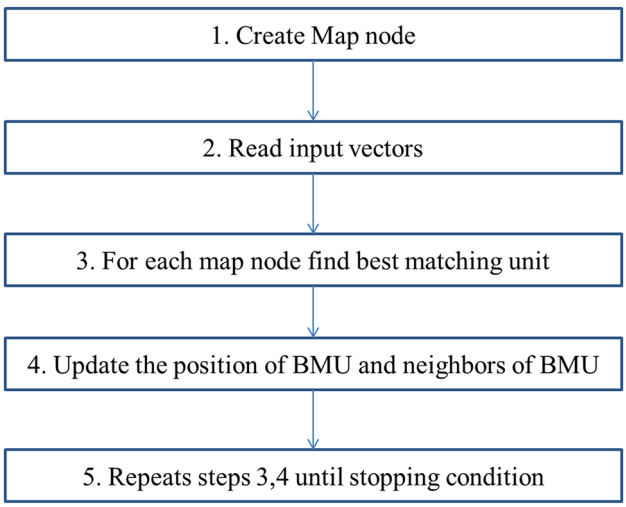

Fig. 2. Working of self-organizing maps.

index of a neighbor of BMU, $\alpha(s)$ is the learning rate, $\Theta(u, v, s)$ is the restraint due to distance from BMU normally called neighborhood function and $D(t)$ is the target output vector.

$$W_v(s+1) = W_v(s).\Theta(u, v, s).\alpha(s).(D(t) - W_v(s)) \quad (8)$$

3.5 *Summary generation*

This step can be divided into two. Intra cluster similarity elimination and summary extraction. They are explained below.

3.5a *Intra cluster similarity elimination:*
The sentences which are so much similar will be removed in this step because, if a sentence conveys the same meaning of another then no need to retain both in the context. The algorithm for similarity elimination is given in Algorithm 1.

Algorithm 1: Intra cluster similarity elimination

Input: Clustered sentences

Output: Reduced clusters

1. Start
2. for each cluster **C**
3. min_dist$_c$ = **α** * far_dist$_c$;
4. for each cluster C
5. for each sentence $\mathbf{S^i_c}$
6. for each sentence $\mathbf{S^j_c}$
7. if (dist($\mathbf{S^i_c}$,$\mathbf{S^j_c}$) <min_dist$_c$)
8. if($\mathbf{\rho(S^i_c)}$> $\mathbf{\rho(S^j_c)}$)
9. remove($\mathbf{S^j_c}$)

else

10. remove($\mathbf{S^i_c}$)
11. end

Now we are left with a n-dimensional vector space contains sentences in the vectorized form and further distance calculation is in Euclidean plane straight line vector distance. Since sentences are vectorized, both sentence vector and sentence were used interchangeably to denote a sentence in this document. A cluster of sentences is the intracluster similarity elimination algorithm is given in Algorithm 1. Each sentence in the cluster is leveled on the basis of the descending order of the relevance factor **ρ**. According to this level assignment, the most relevant sentence in the cluster will have level 1, the next relevant sentence at level 2 position and so on. Consider that there are **m** clusters and each of them with $\mathbf{q_m}$ number of levels. The purpose of similarity elimination is satisfied with the following procedures for each cluster. First, a measurement called far_dist$_c$ calculation is done. It is based on the concept of finding a maximum deviated pair of sentences in the cluster. Generally speaking far_dist$_c$ is the distance of level 1 and level $\mathbf{q_c}$ elements in a cluster. Measurement min_dist$_c$ is actually a benchmark distance because any two sentences, closer than this considered to be redundant. It is actually a fraction of far_dist$_c$ and which is controlled by a similarity factor **α,** whose value ranges from 0 to 1. For each sentence vector's distance is compared with all other $\mathbf{q_c}$-1elements in the cluster. If any pair of sentences has distances less than the corresponding min_dist$_c$ then, one of these sentences must be eliminated. This elimination is based on relevance. The sentences with lesser **ρ** value will be eliminated. This will generate a reduction in the number of sentences in clusters which are having different meanings.

3.5b *Summary generation*:
The algorithm for summary generation is given below.

Algorithm 2: Summary generation

Input: Reduced clusters

Output: Text summary

1. Start
2. for each sentence in level i
3. $\{\mathbf{S^{i,j}_c}\}$ = order($\mathbf{S^i_c}$)
4. for each cluster priority no **j**
5. if(out_count<= **β*** num_sentence)
6. out_sentences{} ←$\mathbf{S^{i,j}_c}$
7. out_count = out_count +1
8. end

From the reduced clusters containing varieties of meaningful sentences, the final summary is generated based on the relevance factor of sentences. Simply the higher level

sentences from each cluster are selected for the output. An output constraint should be imposed to determine how many sentences should present in the output. For that we select all first-level sentences from each cluster, say n number of $\mathbf{S_c^i}$ sentences. The procedure order () will do this. $\{\mathbf{S_c^{i,j}}\}$ denotes a set of prioritized sentences of the level **j** and **i** is the cluster number.The highest priority sentence $\mathbf{S_c^{i,1}}$ will be added to the output, then $\mathbf{S_c^{i,2}}$ and so on. Each time we will check a condition that whether the number of sentences in the output (out_count) is less than the multiple of output factor **β** and number of sentences (num_sentence). This is to assure that the output must be a fraction of the input. This operation is repeated for level 2, level3 and so on till the output constraint is satisfied.

3.6 *An example*

In this section, we show how an actual document will be processed and the summary will be generated. For that consider the input Malayalam document given in figure 3.

Preprocessing is the first step. As discussed in section 3.1 preprocessing is composed of two steps, namely stemming and stop word removal. Stemming is the process of making root words from the original document. Figure 4 gives an example of the stemming of a sentence.

Stop word removal is based on a stop word list part of the dictionary which specifies the number of stop words. If the words in the list present in the document, then they will be eliminated. Subset of such a stop word list for the input shown in figure 1 is given in table 6.

Next step is the entity recognition. As described in section 3.2, entity recognition is the process of labeling each word in the document as entity words, sub-entity words and not entity words. One such example is given in figure 5.

The next step is the relevance analysis phase as discussed in section 3.3. Here we are using three different scoring strategies. The first one is the sentence entity score (section 3.3a), which is the process of assigning numerical values to each sentence based on the entity of that as given in figure 5. Frequent pattern scoring, which explained in section 3.3b is identified and scoring is assigned, which are given in table 7.

Now apply semantic role labeling on each sentence as discussed in section 3.3c. One of such processing is given in figure 6. Note that the verb of the sentence becomes the predicate of a frame and nouns become the arguments. It is possible to make a number of frames from each sentence.

Table 8 shows relevance analysis of some of the sentences from the input document (the system applies relevance analysis for each and every sentence of input, but for convenience, we are showing for a few sentences). Each scoring value and the overall score are normalized to the range of zero to one.

Every value is normalized to the range of 0 to 1 for better comparison and easiness of ranking purpose, min–max normalization is used for this. Like overall ranking, another quantity that is normalized is the similarity score. According to Eq. (6) similarity score calculates vector distance between a sentence and all other sentences in the vector plane, it would be a huge value, but for comparison, we need to reduce the values. For each vector, dimensionality reduction is done. A set of sentences that are vectorized is given in table 9. One question comes is how to determine the dimensions of the vector. They are nothing but repeating words in the document. The repeating words are the words in a frequent pattern as discussed in section 3.3b. This is based on the assumption that, if a word is present in frequent pattern with support count m then that word would also be a frequent pattern with support count greater than or equal to m. In this particular example, we found nine repeating words. Therefore actually the sentence vector has nine dimensions. But for convenience, we are reducing the dimensionality using dimensionality reduction methods, namely missing value ratio [35]. Missing value ratio removes dimensionality whose most of the values are zero. Here threshold was fifty percent of the total number of input sentences. That is, in this particular example, missing value of dimensions greater than seven are reduced, so after removal, only four dimensions exist. With dimensionality reduction, if any of the sentences has zero dimension, i.e. null vectors, then we cannot simply remove those sentences. If their relevance score is high, then we should keep those sentences for future use.

After vectorization of sentences, clustering applied to sentences using SOM as described in section 3.4. The next step is the summary generation as discussed in section 3.5. This has two sub-sections, namely intra-cluster similarity elimination and summary generation. Clustered sentences are passed through the phase called intracluster similarity elimination (section 3.5a). To find the similarity among vectors a similarity matrix is created, which is given in figure 7. The matrix compares the similarity among all sentences in the input document. In figure 7 we show the similarity matrix of the sentences given in table 9. The similarity is measured as the Euclidean distance between vectors. Note that the diagonal elements of the matrix are always zero because if we compare a sentence with itself which gives the distance as zero. Here we are considering the lower triangular matrix only because the upper triangular matrix is simply the replication of the same. If the similarity between two sentences is zero, then two sentences are exactly like each other or both of them are null vector. If a sentence is a null vector we cannot simply avoid sentences if its relevance score is high. Also, those sentences convey different information. Method of solving this is replacing those zeros with a high value denoted by ∞.

Now apply Algorithm 1, the intra-cluster similarity elimination. In this particular example, the min_dist is zero in each cluster. So we have to eliminate one of the sentences from the pair whose similarity value is zero. Such pairs are (S_{13}, S_3), (S_{13}, S_4), and (S_1, S_6). While taking the

ഇന്ത്യയിൽ കേരളസംസ്ഥാനത്തിലുംലക്ഷദ്വീപിലുംപുതുച്ചേരിയുടെഭാഗമായമയ്യഴിയിലുംസംസാരിക്കപ്പെടുന്നഭാഷയാണ്മലയാളം .ഇതുദ്രാവിഡഭാഷാകുടുംബത്തിൽപ്പെടുന്നു.ഇന്ത്യയിൽശ്രേഷ്ഠഭാഷാപദവിലഭിക്കുന്നഅഞ്ചാമത്തെഭാഷയാണ്മലയാളം.ഇന്ത്യൻഭരണഘടനയിലെഎട്ടാംഷെഡ്യൂളിൽഉൾപ്പെടുത്തിയിരിക്കുന്നഇന്ത്യയിലെഇരുപത്തിരണ്ട്ഔദ്യോഗികഭാഷകളിൽഒന്നാണ്മലയാളം.മലയാളഭാഷകൈരളിഎന്നുംഅറിയപ്പെടുന്നു.കേരളസംസ്ഥാനത്തിലെഭരണഭാഷയുംകൂടിയാണ് മലയാളം . കേരളത്തിനുംലക്ഷദ്വീപിനുംപുറമേഗൾഫ്രാജ്യങ്ങൾ,സിംഗപ്പൂർ,മലേഷ്യഎന്നിവിടങ്ങളിലെകേരളീയപൈതൃകമുള്ളഅനേകംജനങ്ങളുംമലയാളംഉപയോഗിച്ചുപോരുന്നു.

ദേശീയഭാഷയായിഉൾപ്പെടുത്തിയത്മറ്റ്21ഭാഷകളുടേതുപോലെതനതായവ്യക്തിത്വംഉള്ളതിനാലാണ്.മലയാളഭാഷയുടെഉല്പത്തിയുംപ്രാചീനതയുംസംബന്ധിച്ചകാര്യങ്ങൾഇന്നുംഅവ്യക്തമാണ് .പഴയതമിഴ്ആണ്മലയാളത്തിന്റെആദ്യരൂപംഎന്നുകരുതുന്നു . യുഎഇയിലെനാലുഔദ്യോഗികഭാഷകളിൽഒന്നുമലയാളംആണു.മലയാളംസംസാരിക്കുന്നജനവിഭാഗത്തിനെപൊതുവായിമലയാളികൾഎന്നുവിളിക്കുമ്പോഴും,ഭാഷയുടെകേരളീയപാരമ്പര്യംപരിഗണിച്ച്കേരളീയർഎന്നുംവിളിച്ചുപോരുന്നു.ലോകത്താകമാനം3.75 കോടിജനങ്ങൾമലയാളഭാഷസംസാരിക്കുന്നുണ്ട്.

ദ്രാവിഡഭാഷാകുടുംബത്തിൽഉൾപ്പെടുന്നമലയാളത്തിനു്.ഇതരഭാരതീയഭാഷകളായസംസ്കൃതം,തമിഴ്എന്നീഉദാത്തഭാഷകളുമായിപ്രകടമായബന്ധമുണ്ട്.സംഘകാലകൃതികളടക്കംഏട്ടാംനൂറ്റാണ്ടുവരെയുള്ളദ്രാവിഡസാഹിത്യംമലയാളത്തിനുകൂടിഅവകാശപ്പെട്ടപൊതുസ്വത്താണ്.സംഘകാലകൃതികളിൽപ്രധാനപ്പെട്ടവപലതുംകേരളത്തിലുണ്ടായതാണെന്ന്കണ്ടെത്തിയിട്ടുണ്ട്..അമ്പതോളംസംഘകാലഎഴുത്തുകാർകേരളീയരായിരുന്നുവത്രെ.ഇപ്പോഴുംപ്രയോഗത്തിലുള്ള150ലധികംമലയാളംവാക്കുകൾസംഘകാലകൃതികളിൽനിന്ന്കണ്ടെത്തിയിട്ടുണ്ട്.

Fig. 3. Input text.

Original sentence:ഇന്ത്യയിൽ

കേരളസംസ്ഥാനത്തിലുംലക്ഷദ്വീപിലുംപുതുച്ചേരിയുടെഭാഗമായമയ്യഴിയിലുംസംസാരിക്കപ്പെടുന്നഭാഷയാണ്മലയാളം.

After stemming: ഇന്ത്യകേരളംസംസ്ഥാനംലക്ഷദ്വീപ്പുതുച്ചേരിഭാഗംമയ്യഴിസംസാരംഭാഷമലയാളം.

Fig. 4. Stemming of the sentence.

Table 6. Sample stop word list.

ഇതു, എന്നും/എന്ന/എന്നീ, മറ്റ്, പോലെ, ആണ്, പോരുന്നു, അതുകൊണ്ട് ,ആ, അല്ല, ഈ, നിന്ന്, കൂടി

ഇന്ത്യ_SNTകേരളം_SNTസംസ്ഥാനം_SNTലക്ഷദ്വീപ്_SNTപുതുച്ചേരി_SNTഭാഗം_NTNമയ്യഴി_SNTസംസാരം_NTNഭാഷ_ETNമലയാളം_ETN.ദ്രാവിഡ_ENTഭാഷാ_ENTകുടുംബം_NTN . ഇന്ത്യ_SNTശ്രേഷ്ഠഭാഷ_ENTപദവി_NTNലഭിക്കുക_NTNഅഞ്ചാം_NTNഭാഷ_ENTമലയാളം_ENT.

Fig. 5. Entity recognition.

Table 7. Frequent patterns from the input.

Pattern	Pattern length	Pattern score
ഭാഷ,മലയാളം	2	0.2
കേരളം, സംസ്ഥാനം, മലയാളം	3	0.3
ഇന്ത്യ, ഭാഷ, മലയാളം	3	0.3
സംഘകാലം,മലയാളം	2	0.2
കേരളം,ഭാഷ,മലയാളം	3	0.3
തമിഴ്,മലയാളം	2	0.2
ദ്രാവിഡ,മലയാളം	2	0.2
തമിഴ്,മലയാളം,ഭാഷ	3	0.3
ഔദ്യോഗിക,ഭാഷ	2	0.2

first pair we are determining which sentence to be eliminated. It is based on the relevance value of the individual sentence. By looking into table.8, it is clear that the sentence S_3 is more relevant than S_{13}. So eliminate S_{13}. Now pair 2 needs not to be processed because S_{13} is already eliminated. Consider pair 3, out of which S_1 and S_6, S_1 is more relevant than S_6. So eliminate S_6. After the application of Algorithm 1, the cluster assignment of sentences is given in figure 8. The suffix of S represents the cluster number to which the sentence belongs and the superscript denotes the sentence number. In bracket, the normalized relevance value is given.

Each cluster is leveled based on a descending order of relevance. In cluster1 $\mathbf{S_1^6}$ is in level 1 position and $\mathbf{S_1^7}$ in level 2 position. Each cluster is leveled like this. The advantage is that if you select level 1 you will get the sentences with high relevance in each topic because each cluster contains different topic sentences. The level and the elements of that level are given below.

$$\text{Level 1} = \left\{\mathbf{S_1^6}, \mathbf{S_2^1}, \mathbf{S_3^2}\right\} \quad \text{Level 2} = \left\{\mathbf{S_1^7}, \mathbf{S_2^4}, \mathbf{S_3^{14}}\right\}$$
$$\text{Level 3} = \left\{\mathbf{S_3^{11}}\right\}$$

The next process is the assignment of priority values. The below set gives ordered pairs of sentences without priority number and with priority number. Priority numbering is nothing but within the sentences of a particular level again ordering them in the descending order of relevance score. The new notation adds an extra element in the superscript which is the priority number.

$$\text{Level 1} = \left\{\left(\mathbf{S_3^3}, \mathbf{S_3^{3,1}}\right), \left(\mathbf{S_2^1}, \mathbf{S_2^{1,2}}\right), (\mathbf{S_1^6}, \mathbf{S_1^{6,3}})\right\},$$
$$\text{Level 2} = \left\{\left(\mathbf{S_3^{14}}, \mathbf{S_3^{14,1}}\right), \left(\mathbf{S_2^4}, \mathbf{S_2^{4,2}}\right), (\mathbf{S_1^7}, \mathbf{S_1^{7,3}})\right\},$$
$$\text{Level 3} = \left\{(\mathbf{S_1^{11}}), (\mathbf{S_1^{11,1}})\right\}$$

By applying summary generation algorithm discussed in section 3.5b, the summary is generated and is given in figure 9.

4. Results and discussions

4.1 *Theoretical analysis*

This section contains a number of theoretical evaluations carried out in the various steps of the system. Note that theoretical analysis means evaluating the system theoretically. Here we perform the gradient calculation and maximization of relevance equation and complexity analysis of the algorithms that we were developing.

4.1a *Gradient calculation of relevance equation*
Equation (9) gives the relevance calculation of a sentence. From the equation, it is clear that the relevance of a sentence is calculated by three features, namely: sentence_entity_score, frequent_pattern_score, and similarity_score. Our aim is to find which one of these features is more influencing in the relevance equation. Once we find such a feature out of three then we can do optimizations on that feature in order to increase the output. Gradients are used for that purpose. Gradient calculation needs a function with variables. Here relevance **ρ** can be represented as a function of three above features. So **ρ** is a function of three variables, namely x, y and z corresponding to three features. Now **ρ** is represented as $\mathbf{ax + by + cz}$ where x, y, z are features and the constants a, b, c are the weight assignment of the features. Weight assignment is needed when we want to assign different priorities to features. Assigning a different weight to each feature is a system designing choice. If the designer thinks the semantic similarity should have given importance than lexical similarity features, then similarity_score will get more

S1:ഇന്ത്യയിൽ

കേരളസംസ്ഥാനത്തിലുംലക്ഷദ്വീപിലുംപുതുച്ചേരിയുടെഭാഗമായമയ്യഴിയിലുംസംസാരിക്കപ്പെടുന്നഭാഷയാണ്മലയാളം.

Φ_1: <സംസാരിക്കപ്പെടുന്ന, ഇന്ത്യ[AM-LOC],കേരളസംസ്ഥാനം[AM- LOC], ലക്ഷദ്വീപ്[AM- LOC],പുതുച്ചേരി[AM- LOC],മയ്യഴി[AM- LOC]>

Φ_2:<സംസാരിക്കപ്പെടുന്ന,ഭാഷ[AM- THEME],മലയാളം[AM- THEME]>

Fig. 6. Sample semantic frame creation.

Table 8. Scoring of sentences.

Sentence no	Sentence entity scores	Frequent pattern score	Similarity scores	Overall score $\rho(S_i)$
S1	0.097	0.098	0.070	0.083
S3	0.071	0.098	0.111	0.093
S4	0.063	0.137	0.091	0.082
S6	0.063	0.058	0.072	0.067
S7	0.089	0	0.046	0.006
S11	0.052	0.039	0.072	0.006
S12	0.072	0.058	0.046	0.058
S13	0.040	0.098	0.067	0.058
S14	0.100	0.078	0.113	0.105

Table 9. Vectorized sentences.

Sentence no	Vector
S_1	[1 1 1 1]
S_3	[0 1 1 1]
S_4	[0 1 2 1]
S_6	[1 1 0 1]
S_7	[2 1 0 0]
S_{11}	[0 1 0 1]
S_{12}	[1 1 0 1]
S_{13}	[0 1 0 1]
S_{14}	[0 1 0 2]

importance and according to that c will be assigned a higher value than a or b.

To carry out the gradient calculation, we need to know the value of x, y, z at an arbitrary point. Here we select a random sentence from the input and finding feature values of the same. So for a sentence feature vector is$(0.6, 0.1, 0.3)$. Gradient calculation represents the output function as given below.

$$\boldsymbol{\rho} = \mathbf{ax} + \mathbf{by} + \mathbf{czat}(\mathbf{x}, \mathbf{y}, \mathbf{z}) = (0.6, 0.1, 0.3) \qquad (9)$$

Now performing partial differentiation on $\boldsymbol{\rho}$ with respect to x, y, z.

$$\nabla\boldsymbol{\rho} = \left(\frac{\partial\boldsymbol{\rho}}{\partial\mathbf{x}}, \frac{\partial\boldsymbol{\rho}}{\partial\mathbf{y}}, \frac{\partial\boldsymbol{\rho}}{\partial\mathbf{z}}\right)$$

We will get the partial derivatives of $\boldsymbol{\rho}$ with respect to x, y, z as a, b, c respectively.

$$\frac{\partial\boldsymbol{\rho}}{\partial\mathbf{x}} = \mathbf{a}\frac{\partial\boldsymbol{\rho}}{\partial\mathbf{y}} = \mathbf{b}\frac{\partial\boldsymbol{\rho}}{\partial\mathbf{z}} = \mathbf{c}$$

It is represented as:

$$\nabla\boldsymbol{\rho}(0.6, 0.1, 0.3) = (\mathbf{a}, \mathbf{b}, \mathbf{c})$$
$$\nabla(\mathbf{ax} + \mathbf{by} + \mathbf{cz})|(\mathbf{x}, \mathbf{y}, \mathbf{z}) = (\mathbf{a}, \mathbf{b}, \mathbf{c})$$

$$\begin{bmatrix}
0 & & & & & & & & & \\
1 & 0 & & & & & & & & \\
1.4 & 1 & 0 & & & & & & & \\
0 & 1 & 1.4 & 0 & & & & & & \\
2 & 2.2 & 2.4 & 2 & 0 & & & & & \\
1 & 1 & 1.7 & 1.4 & \infty & 0 & & & & \\
1 & 1.4 & 1.4 & 1.4 & 1.7 & 1.4 & 0 & & & \\
1 & 1.4 & 1 & 1.4 & 1.4 & 2.4 & 2.2 & 0 & & \\
1 & 0 & 0 & 1 & 1 & 1 & 1.7 & 1.4 & 0 & \\
1.4 & 1.4 & \infty & 1 & 1.4 & 1.4 & \infty & 1.7 & 1.4 & 0
\end{bmatrix}$$

Fig. 7. Similarity matrix.

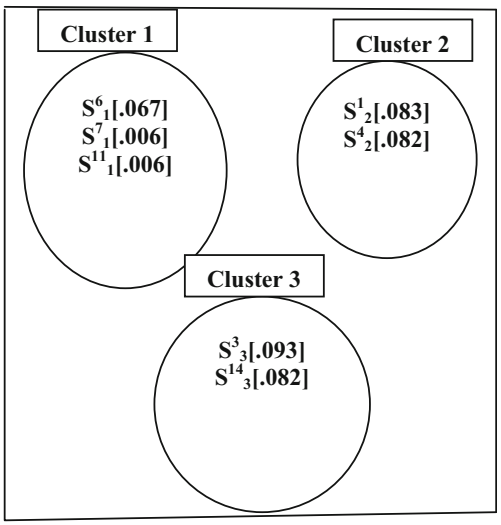

Fig. 8. Cluster organization.

The final output is:

$$\nabla(\mathbf{ax} + \mathbf{by} + \mathbf{cz})|_{(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})=(0.6,0.1,0.3)} = (\mathbf{a}, \mathbf{b}, \mathbf{c}) \qquad (10)$$

The output is known as gradient vector which is (a, b, c). Normally we will put the values of x, y, z in the gradient vector equation, from where we will find the greater valued variable. But in this case, the equation is linear. So while taking partial derivative, function reduces in terms constants. From the evaluation, it is clear that the more influencing feature in the output is based on the weight that we have given to it. Keeping these weights outside, each feature is having equal importance in the output $\boldsymbol{\rho}$.

4.1b *Optimizing relevance equation:*
In the next evaluation, we are going for optimization on the relevance calculation equation given in Eq. (9). The optimization finds the maximum value of the function under given constraints. This is carried out by the method of Lagrange's multiplier. For that purpose, we need a

ഇന്ത്യയിൽ കേരളസംസ്ഥാനത്തിലുംലക്ഷദ്വീപിലുംപുതുച്ചേരിയുടെഭാഗമായമയ്യഴിയിലുംസംസാരിക്കപ്പെടുന്നഭാഷയാണ്മലയാളം .ഇന്ത്യയിൽശ്രേഷ്ഠഭാഷാപദവിലഭിക്കുന്നഅഞ്ചാമത്തെഭാഷയാണ്മലയാളം.ഇന്ത്യൻഭരണഘടനയിലെഎട്ടാംഷെഡ്യൂളിൽഉൾപ്പെടുത്തിയിരിക്കുന്നഇന്ത്യയിലെഇരുപത്തിരണ്ട്ഔദ്യോഗികഭാഷകളിൽഒന്നാണ്മലയാളം.കേരളസംസ്ഥാനത്തിലെഭരണഭാഷയുംകൂടിയാണ് മലയാളം .കേരളത്തിനുംലക്ഷദ്വീപിനുംപുറമേഗൾഫ്രാജ്യങ്ങൾ, സിംഗപ്പൂർ, മലേഷ്യഎന്നിവിടങ്ങളിലെകേരളീയപൈതൃകമുള്ളഅനേകംജനങ്ങളുംമലയാളംഉപയോഗിച്ചുപോരുന്നു.യുഎഇയിലെനാലുഔദ്യോഗികഭാഷകളിൽഒന്നുമലയാളംആണു.ദ്രാവിഡഭാഷാകുടുംബത്തിൽഉൾപ്പെടുന്നമലയാളത്തിനു്, ഇതരഭാരതീയഭാഷകളായസംസ്കൃതം, തമിഴ്എന്നീഉദാത്തഭാഷകളുമായിപ്രകടമായബന്ധമുണ്ട്.

Fig. 9. Summarized text.

constraint to be satisfied. Here the function to be optimized is $\boldsymbol{\rho}$ and the constraint $\mathbf{g}(\mathbf{x},\mathbf{y},\mathbf{z}) = \mathbf{x}+\mathbf{y}+\mathbf{z} = 3$. The constraint g (x, y, z) is nothing but the maximum values possible for each feature. For more understanding, we could take the example given in table.8. It is clear that feature values are normalized into the range of o to 1. That is if you add all the values of a single feature, it will evaluate as 1. It says that each feature value of a particular sentence tends to 1. So it could be represented as in Eq. (11).

$$\lim_{\{x,y,z\}\to 1}(ax+by+cz) \tag{11}$$

Now we have got the constraint $\mathbf{x}+\mathbf{y}+\mathbf{z} = 3$ i.e. the maximum value of x, y, z is 1 then their sum is 3. So the equation for applying Lagrange's multiplier is given in (12).

$$\boldsymbol{\rho} = \mathbf{ax}+\mathbf{by}+\mathbf{cz}\,\text{with constraint}\,\mathbf{x}+\mathbf{y}+\mathbf{z} = 3 \tag{12}$$

Consider the operation that we are partial differentiating the relevance equation and the constraint with respect to x,y and z and equating them.

$$\boldsymbol{\rho} = \mathbf{ax}+\mathbf{by}+\mathbf{cz}, \mathbf{g} = \mathbf{x}+\mathbf{y}+\mathbf{z} = 3$$

Partial is differentiated with respect to x:

$$\frac{\partial \boldsymbol{\rho}}{\partial \mathbf{x}} = \mathbf{a}\frac{\partial \mathbf{g}}{\partial \mathbf{x}} = 1\,\text{on equating we get}\,\mathbf{a} = 1$$

Partial is differentiated with respect to y:

$$\frac{\partial \boldsymbol{\rho}}{\partial \mathbf{y}} = \mathbf{b}\frac{\partial \mathbf{g}}{\partial \mathbf{y}} = 1\,\mathbf{on}\,\text{equating we get}\;\;\mathbf{b} = 1$$

Partial is differentiated with respect to z:

$$\frac{\partial \boldsymbol{\rho}}{\partial \mathbf{z}} = \mathbf{c}\frac{\partial \mathbf{g}}{\partial \mathbf{z}} = 1\,\text{on equating we get}\;\;\mathbf{c} = 1$$

The output of Lagrange's multiplier is described as three cases. Cases are nothing but the maximum value obtained by the function of certain conditions. Conditions are the possible values of constants in the equation. It gives the prediction of the output when constants a, b, c have extreme values like 0 or 1 or equality among constants. The function achieves minimum value when all of its constants a, b, c are zero and obtains maximum value when these constants are equal. Here those conditions are given as three cases. Each case describes the output of the function as a 0 (minimum), 3c (maximum) and infinity.

Case1: The value of the function is 0 if the entire variable values a, b, c are zero. Output value is 3c for a=c and if the following constraint is satisfied. Otherwise the value is infinity.

$$\mathbf{max}\{\mathbf{ax}+\mathbf{by}+\mathbf{cx}|\mathbf{x}+\mathbf{y}+\mathbf{z} = 3\}$$
$$= \begin{cases} \mathbf{0}, & (\mathbf{c} = 0 \wedge \mathbf{b} = 0 \wedge \mathbf{a} = 0) \\ & (\mathbf{c} > 0 \wedge \mathbf{b} = \mathbf{c} \wedge \mathbf{a} = \mathbf{c}) \vee \\ \mathbf{3c}, & (\mathbf{c} < 0 \wedge \mathbf{b} = \mathbf{c} \wedge \mathbf{a} = \mathbf{c}) \\ & \mathbf{for\ a} = \mathbf{c} \\ \infty, & \mathbf{otherwise} \end{cases}$$

Case2: The value of the function is 0 if the entire variable values a, b, c are zero. Output value is 3c for b=c and if the following constraint is satisfied. Otherwise the value is infinity.

$$\mathbf{max}\{\mathbf{ax}+\mathbf{by}+\mathbf{cx}|\mathbf{x}+\mathbf{y}+\mathbf{z} = 3\}$$
$$= \begin{cases} \mathbf{0}, & (\mathbf{c} = 0 \wedge \mathbf{b} = 0 \wedge \mathbf{a} = 0) \\ & (\mathbf{c} > 0 \wedge \mathbf{b} = \mathbf{c} \wedge \mathbf{a} = \mathbf{c}) \vee \\ \mathbf{3c}, & (\mathbf{c} < 0 \wedge \mathbf{b} = \mathbf{c} \wedge \mathbf{a} = \mathbf{c}) \\ & \mathbf{for\ a} = \mathbf{c} \\ \infty, & \mathbf{otherwise} \end{cases}$$

Case3: The value of the function is 0 if the entire variable values a, b, c are zero. Output value is 3c for $\mathbf{x} = -\mathbf{y}-\mathbf{z}+3$ and if the following constraint is satisfied. Otherwise the value is infinity.

$$\mathbf{max}\{\mathbf{ax}+\mathbf{by}+\mathbf{cx}|\mathbf{x}+\mathbf{y}+\mathbf{z} = 3\}$$
$$= \begin{cases} \mathbf{0}, & (\mathbf{c} = 0 \wedge \mathbf{b} = 0 \wedge \mathbf{a} = 0) \\ & (\mathbf{c} > 0 \wedge \mathbf{b} = \mathbf{c} \wedge \mathbf{a} = \mathbf{c}) \vee \\ \mathbf{3c}, & (\mathbf{c} < 0 \wedge \mathbf{b} = \mathbf{c} \wedge \mathbf{a} = \mathbf{c}) \\ & \mathbf{for\ x} = -\mathbf{y}-\mathbf{z}+\mathbf{3} \\ \infty, & \mathbf{otherwise} \end{cases}$$

From the evaluation it is clear that the maximum value of the function strictly depends on the values of the weight factors a, b, c. This property achieved by the equation because, it is linear in nature.

4.1c *Complexity calculation of algorithms:*
Here algorithmic complexity is calculated for the algorithms proposed in this paper. First of all, takes Algorithm 1 into consideration, then Algorithm 2.

Complexity of Algorithm 1: For Algorithm 1 there are two iterations. One is a simple iteration and another one is a nested iteration. Consider we have n number of sentences and m number of clusters. The first iteration is used for the calculation of far_dist$_c$. It will find the farthest sentences in the vector plane. While mapping n number of sentences into m clusters, one cluster would have at most $\mathbf{n/m}$ number of sentences. So to find a far_distc maximum number of calculations in the worst case is:

$$\textbf{number of far\_distance calculation} = \frac{\frac{\mathbf{n}}{\mathbf{m}}\left(\frac{\mathbf{n}}{\mathbf{m}}-1\right)}{2}$$

This is approximated as:

$$\mathbf{O}\left(\frac{\mathbf{n}^2}{\mathbf{m}^2}\right)$$

The iteration works for each cluster. It works m times. So the total complexity iteration 1 is:

$$\mathbf{mx}\left(\frac{\mathbf{n}^2}{\mathbf{m}^2}\right) = \frac{\mathbf{n}^2}{\mathbf{m}}$$

Denoted as:

$$\textbf{Loop}1\ \textbf{complexity} = \mathbf{O}\left(\frac{\mathbf{n}^2}{\mathbf{m}}\right) \tag{13}$$

While considering the loop 2 the nested loop, the outer loop works for each cluster i.e. it works for m times.

$$\textbf{Loop}\ 2\ \textbf{outer loop complexity} = \mathbf{O}(\mathbf{m})$$

The both inner loops work on the sentences in each cluster. That is, both of them works $\mathbf{n/m}$ times. So the complexity of inner loops of loop 2 is:

$$\textbf{Loop}\ 2\ \textbf{inner loops complexity} = \mathbf{O}\left(\frac{\mathbf{n}^2}{\mathbf{m}^2}\right)$$

Hence the loop2 complexity is:

$$\begin{aligned}\mathbf{mx}\left(\frac{\mathbf{n}^2}{\mathbf{m}^2}\right) &= \frac{\mathbf{n}^2}{\mathbf{m}}\\ \textbf{Loop}\ 2\ \textbf{complexity} &= \mathbf{O}\left(\frac{\mathbf{n}^2}{\mathbf{m}}\right)\end{aligned} \tag{14}$$

The total complexity of Algorithm 1 is the sum of loop 1 complexity and loop 2 complexity. That is, adding Eqs. (13) and (14).

$$\mathbf{O}\left(\frac{\mathbf{n}^2}{\mathbf{m}}+\frac{\mathbf{n}^2}{\mathbf{m}}\right) \approx \mathbf{O}\left(\frac{\mathbf{n}^2}{\mathbf{m}}\right)$$

Hence the complexity of Algorithm 1 is:

$$\textbf{Algorithm}\ 1\ \textbf{complexity} = \mathbf{O}\left(\mathbf{n}^2\right)$$

Complexity of Algorithm 2:
Algorithm 2 is for generating output summary. The sentences in each cluster are organized into different levels and sentences in each level are again organized to priority numbers in the descending order of relevance. Algorithm 2 contains one nested loop. The outer loop works for each level and the inner loop works for priority number. Consider that we have **a** number of levels in each cluster and **b** number of sentences in each level. The out loop works for '**a**' times and inner loop works for '**b**' times. So total complexity is:

$$\textbf{nested loop complexity} = \mathbf{O}(\mathbf{a}.\mathbf{b})$$

One thing to be noted is that inside the loop is the IF condition. It says that the loop operation should be terminated when the number of output sentences is a fraction of the number of input sentences. For example, if we chose $\boldsymbol{\beta} = 0.2$ and the number of sentences in the input is 100 then the number of output sentences would be$0.2\mathbf{X}100 = 20$. That is, in output we would get 20 sentences. We could summarize it in such a way that whatever may be the value of **a** and b the nested loop will work for a fraction of **n** times. So we can equate the as:

$$\mathbf{a}.\mathbf{b} = \boldsymbol{\beta}.\mathbf{n}$$

From this the complexity of the algorithm is evaluated as:

$$\mathbf{O}(\mathbf{a}.\mathbf{b}) = \mathbf{O}(\boldsymbol{\beta}.\mathbf{n})$$

It is approximated as:

$$\mathbf{O}(\boldsymbol{\beta}.\mathbf{n}) \approx \mathbf{O}(\mathbf{n})$$

The complexity of Algorithm 2 is:

$$\textbf{Algorithm}\ 2\ \textbf{complexity} = \mathbf{O}(\mathbf{n})$$

4.2 *Experiments and results*

The system efficiency is calculated by comparing with a number of existing systems as well as making available of some standard test methods prevailing in computational linguistics. The first one is comparing the system with existing online summarizers; next is comparing the system with an existing offline system, finding the efficiency of the system with and without clustering methods. The standard tests such as question game evaluation, sentence rank evaluation, and keyword association are also applied. For

all evaluations, datasets selected from popular Malayalam websites such as 'Manoramaonline.com' [36], 'Wikipedia' and meaning of sentences and other preprocessing tasks carried out with the help of Malayalam online dictionary 'Olam'. Each data set is the collection of articles about some random topics in Malayalam script. So that we can view each data set as a chunk of documents with more than three paragraphs and they are on different topics (includes sports, politics, weather, film, science news and editorials), different sizes (ranges from 200-1000 words). There are five datasets we have taken with each of them having 10 to 15 documents. The performance measure of each data set is the average of performance values of documents present in them.

Experimental setup for SOM: In order to train SOM network five data sets are selected as described above. The vectorized notation of sentences from each dataset is taken as the training vector. The initial weight matrix contains randomly chosen values in the range 0 to 1, and further tuning of parameters is done by pieces of training with examples. Sensitivity analysis is carried out in order to determine the performance of the algorithm with the variation of parameters as well as the determination of most influencing dimensions. Weight sensitivity is calculated by ranking the weights obtained in iterations. As explained in section 3.6, every training sample is a vectorized sentence in which dimensions are the most frequent words of the document. For each document in six data sets, SOM is performed and weights obtained for each feature are calculated and presented in descending order. For every sample, the feature with a higher weight is given as a scatter plot in figure 10. The x-axis denotes the documents which were used for training and the y-axis denotes the features involved in vectorization of sentences. For uniqueness instead of putting a specific number of features, they are represented by percentage notation. For example, a 10 percentage feature means it is the feature at 1/10th (flooring operation is applied in the case of decimal number) dimension of the vector. Plot denotes the feature with which the highest weight is associated after training. The scatter plot is heavily concentrated on the bottom portion, which infers that the first few features of the sentences are always being obtained high weight. This is because of the arrangement of features is in such a way that the most repeating word became the first dimension and second-highest become the next dimension and so on. It substantiates that frequent words having higher weights so that they are more influential in clustering. The experiment carried out in a laptop equipped with an intel core i3-6006U processor, 4GB of RAM and built-in Intel®HDGraphics 530.

4.2a *Comparison with online summarizers:*
The comparison with online summarizers is given in table.10. Here we have selected the online summarizers: Textsummarization.net, Textcompactor.net, Summarygenerator.com, and Splitbrain.org. All these four summarizers compared with the proposed methods. Precision, recall, and F-measures were found. Precision [37] is the ratio calculated as the intersection between the expected number of sentences and retrieved sentences divided by the number of retrieved sentences. The recall is the ratio between intersections between expected numbers of sentences and retrieved sentences divided by the expected number of sentences. The harmonic mean of precision and recall is called an F-measure.

The online summaries are Text summarization, Textcompactor, Summarygenerator, and Splitbrain. Table 10 gives a comparison score. From the calculations, it is clear that the proposed method is far better than the existing offline summarizer. It is because of the use of strong sentence scoring features. The three-sentence scoring features taken here in this approach is sentenced entity score, similarity score, and frequent pattern score. Instead of applying simple mathematical equations, they score sentences based on the content of the sentence. For example, sentence entity score will score the sentence only if the entity words are present in the sentence. It implies that the entity words are so important for the document and the sentences contain the entity words are also important. The similarity score is not a simple common word calculation which lies in syntax level. Here we apply semantic role labeling which will consider the meaning of the sentence also. Frequent patterns are actually some series of words which are occurring more times in the document. We are implementing this feature in the system with a view that if some words are frequent then they must have some importance in the document.

4.2b *Comparison with existing offline summarizer:*
A comparison with an offline system is also necessary for evaluation. For that, we have selected an extractive Malayalam text summarizer developed by ourselves [6]. It works based on the concept of minimal spanning tree-based summarization. Here the graph is plotted (figure 11) based on the F-measure [38], the corresponding values are given in table 11. In the existing system, the only relevance scoring is present and it does not deal with the similarity elimination whereas the performance is increased in the proposed system is because of the implementation of the similarity elimination algorithm. In effect, redundancy is not at all a concern of the existing system, but the proposed system cares both redundancy and relevance.

4.2c *Comparison with and without similarity elimination phase:*
We want to measure the efficiency of the clustering algorithm and the intracluster similarity elimination algorithm that we proposed in this paper. These two steps are combined into a single stage called the similarity elimination phase. For that, we are performing a sentence similarity test on the input document and the output document. It verifies whether redundancy is reduced after applying summarization or not. Two types of outputs are taken; firstly we are

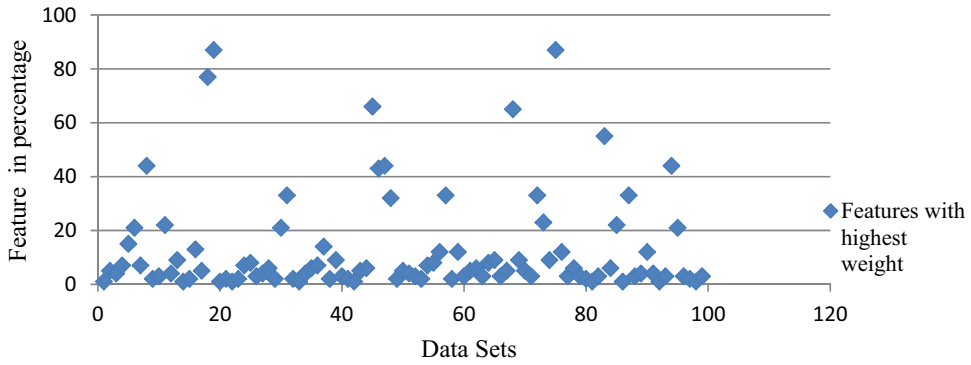

Fig. 10. Sensitivity analysis of SOM.

Table 10. Comparison with online systems.

System	Precision	Recall	F-measure
Proposed method	.8	.667	.738
Textsumarization (http://textsummarization.net/text-summarizer)	.250	.316	.240
Textcompactor (https://www.textcompactor.com/)	.483	.345	.398
Summarygenerator (http://summarygenerator.com/)	.562	.304	.384
Splitbrain (https://www.splitbrain.org/services/ots)	.499	.314	.3725

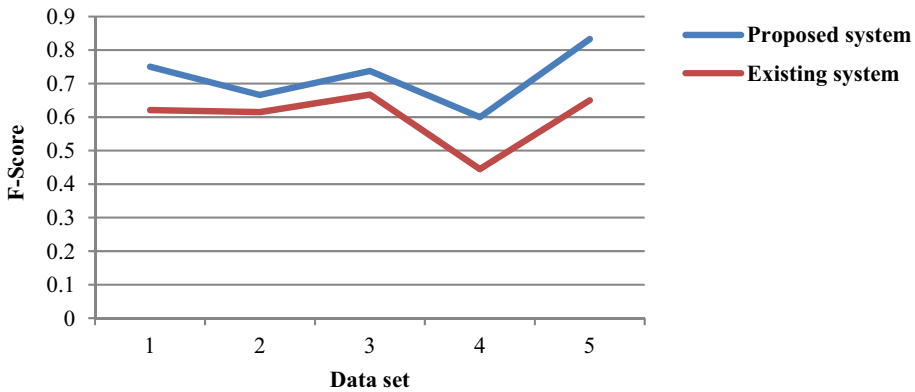

Fig. 11. Comparison with an existing system.

taking the output after relevance analysis phase i.e. top-ranked sentences will be in the output. Later the output is taken at the end of the system working steps i.e. the output with the clustering phase. The F-score measure is used to find efficiency. The output is given in figure 12 and the corresponding F-Score values are given in Table.12. Hence it is proven that the system had improved performance with clustering and intracluster similarity elimination than without similarity elimination phase.

4.2d *Question game evaluation:*
A type of extrinsic summary evaluation method is called a question game [37]. The extrinsic method evaluates the extent of accessibility of the result, in other words, it measures the capability of the output to maintain the constraints and rules proposed by the system. The question game starts by giving the input document to a human reader. After reading tester asks a few questions about the topics to the reader. The reader answers these questions three times. They are: Without looking into the passage (called baseline 1), by looking into the document (baseline 2) and by looking into the summary. A summary is effective if it contains most of the answers for the questions and also the summary should be closer to baseline1 not to baseline 2. One such question game experiment is with datasets are given in figure 13 and corresponding values are

Table 11. Comparison with existing system.

Data set	F-Score	
	Proposed system	Existing system
1	0.75	0.62
2	0.66	0.61
3	0.73	0.66
4	0.60	0.44
5	0.83	0.65

given in table 13. Datasets are on the X-axis and the percentage of questions answered is on the Y-axis. From the figure, it is clear that the summary is always near to baseline 1 not to baseline 2. That is because the baseline 1 is the answers said by the person from his memory. The points a person memorizes after reading a document is actually the human-generated summary of that particular document. If the system generated summary is more related to the human-generated summary, we could say that the system has good performance. Here the system has also given good results because of the reason that they are closer to baseline 1 not to baseline 2.

4.2e *Sentence rank evaluation:*
Sentence Rank [38] is a better method of finding the information content of the summary. In this, sentences are ranked by their relevance. A reference summary is generated manually based on the relevance of these sentence ranking. After that, the system generated summary is being compared with the reference summary. The graph is plotted in figure 14 with data sets in the X-axis and Y-axis contains the sum of relevance score of each document for reference summary, and system-generated summary, which are normalized to the range of 0 to 100. The relevance scores are also given in table 14. One advantage of evaluating the system with sentence rank evaluation is that it compares the reference summary and system-generated summary. From figure 13 it is clear that the relevance score of the system generated summary and reference summary is almost similar. It says that the system works in such a way that the developer designs it to work, which is the desirable property. So we can say that our system is functioning in such a way that as we expected it to work.

4.2f *Key word association:*
In key word association [38] test, there is a key word list extracted from the original input document. In the summary we are finding the percentage of the keywords associated with it. More percentage value means the summary included more topics of the input. Such an experiment is given in figure 15 and corresponding values are given in table 15. Datasets are in the X-axis and the percentage of keywords included in the output is in Y-axis. From the analysis, it is clear that there exists a large fraction of keywords in the output summary. The high rate of keywords is because of the application of sentence entity score. In our method we are categorizing words into main entity words, sub-entity words and not-entity words. Out of these, main entity words are getting more priority, so that there exists a probability of selecting those sentences which have more number of main entity words. These main entity words are actually the keywords in the keyword association technique. In short, we can say that our system got higher performance in keyword association testing due to the implementation of entity recognition.

5. Conclusion and future work

This paper proposes a method for extractive summarizer in Malayalam. From the evaluation, it is evident that the system will work better than the existing systems. Implementation of entity recognition is a notion of presenting sentences that contain more important words. The relevance analysis contains three strong sentences scoring features that are scoring the sentences based on the content rather than applying some rough mathematical operations.

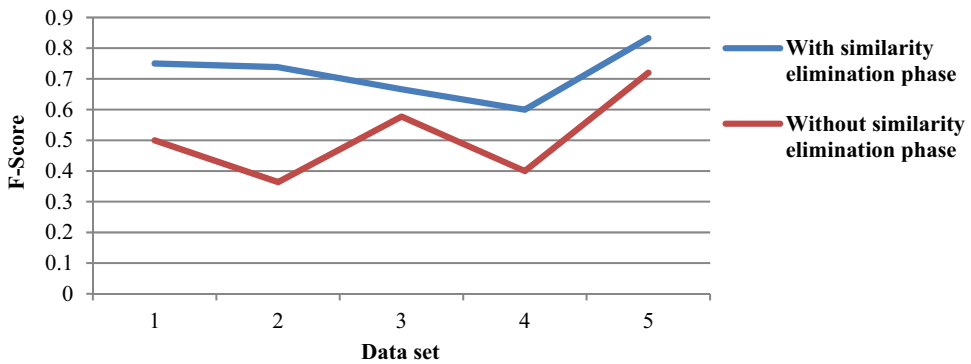

Fig. 12. Comparison of with and without clustering methods.

Table 12. Comparison with and without similarity elimination phase.

Data set	F-Score	
	With similarity elimination phase	Without similarity elimination phase
1	0.75	0.62
2	0.66	0.61
3	0.73	0.66
4	0.60	0.44
5	0.83	0.65

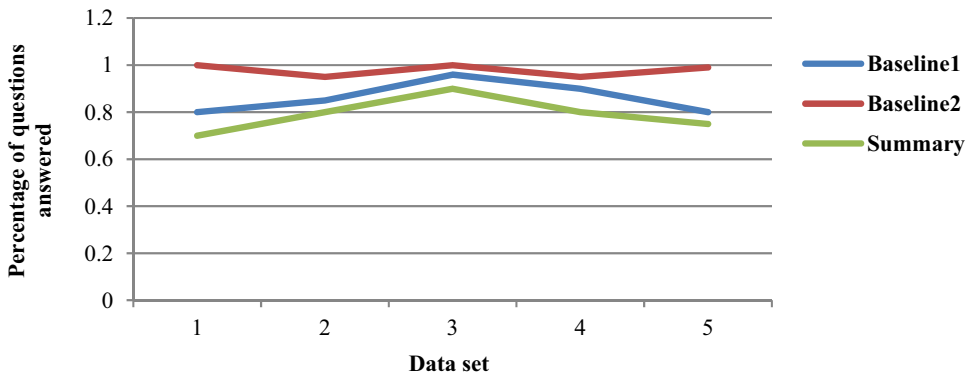

Fig. 13. Question game evaluation.

Table 13. Question game evaluation.

Data set	The percentage of questions answered		
	Baseline 1	Baseline 2	Summary
1	0.80	1.00	0.70
2	0.85	0.95	0.80
3	0.96	1.00	0.90
4	0.90	0.95	0.80
5	0.80	0.99	0.75

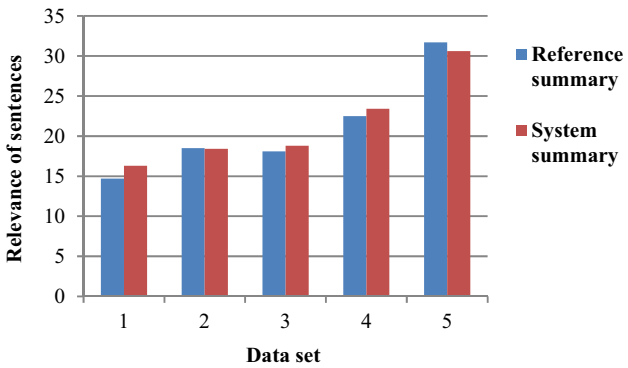

Fig. 14. Sentence rank evaluation.

That is before in scoring calculation we are identifying the topics from the document so that further scoring is based on these topic assessments. The clustering algorithm is actually representing sentences in clusters. So that it provides a notion for similarity elimination and keeping diversity. Hence, by clustering, we actually divided the documents into subtopics. The intracluster similarity elimination reduces redundancy in clusters by eliminating the redundant sentences. The summary generation algorithm preserves both relevance and diversity. The summary generator starts selecting sentences from higher levels. In higher levels, sentences with higher relevance would be present. So we can say that the system generates higher relevant sentences as output. By considering the other property of the summary generation algorithm, it would select the sentences from each cluster. Already we know that each cluster represents subtopics of the document. So the system would generate the summary by giving equal importance to each subtopic, thus preserves diversity. Yet, some enhancements are open for the proposed system. The SRL used here is a shallow SRL for semantic role labeling. In future, use of deep SRL may improve system efficiency. The efficiency of the clustering algorithm can also be improved with some extensions. Here we are performed SRL with three roles as main. In the future, it may extend with more number of roles. The entities of this system can

Table 14. Sentence rank evaluation.

Data set	Relevance score	
	Reference summary	System summary
1	14.7	16.3
2	18.5	18.4
3	18.1	18.8
4	22.5	23.4
5	31.7	30.6

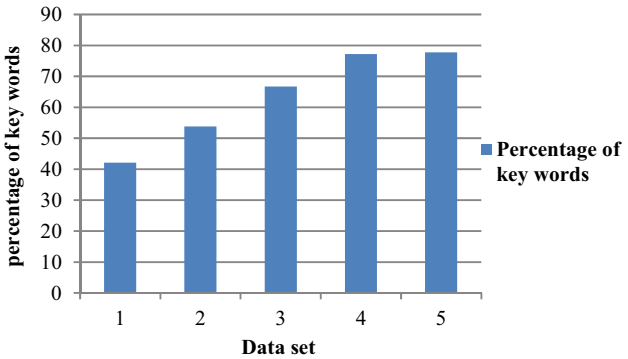

Fig. 15. Key word association evaluation.

Table 15. Key word association evaluation.

Data set	Percentage of key words
1	42.1
2	53.8
3	66.7
4	77.2
5	77.78

also be expanded so as to process the number of topic-based documents.

Acknowledgement

We express our gratitude towards all the faculty members and students of Ilahia college of engineering and technology for their immense support.

References

[1] Ercan C and Igor K 2011 *Semantic role frames graph-based multidocument summarization*, University of Ljubljana, Slovenia Proc. SiKDD'11: 01–11

[2] Renjith S R and Sony P 2015 An automatic text summarization for Malayalam using sentence extraction, In: *IRF International Conference*, 14th June 2015, Kerala

[3] https://kerala.gov.in/official-language-legislative-commission

[4] Ajmal E B and Haroon R P 2015 An extractive Malayalam document summarization based on graph theoretic approach. In: *2015 International Symposium on Web of Things and Big Data (WoTBD 2015)*, 18–20 October, Manama, Bahrain, IEEE 2015

[5] Patil K and Brazdil P 2007 Text summarization: using centrality in the pathfinder network. *Int. J. Comput. Sci. Inform. Syst* [online] 2: 18–32

[6] Rahul Raj M and Haroon R P 2016 *Malayalam text summarization: minimum spanning tree based graph reduction approach*, 978-1-5090-3480-2/16/$31.00 ©2016 IEEE

[7] Gupta V and Lehal G S 2010 A survey of text summarization extractive techniques. *J. Emerg. Technol. Web Intell.* 2: 258–268

[8] Saranyamol C and Sindhu L 2014 A survey on automatic text summarization. *(IJCSIT) Int. J. Comput. Sci. Inf. Technol.* 5: 7889–7893

[9] Kabeer R and Idicula S M 2014 *Text summarization for Malayalam documents – an experience.* IEEE 2014

[10] Ajmal E B and Haroon R P 2016 Maximal marginal relevance based malayalam text summarization with successive thresholds. *Int. J. Cybern. Inform. (IJCI)* 5(2):349–356

[11] Xie S and Liu Y 2008 Using corpus and knowledge-based similarity measure in maximum marginal relevance for meeting summarization. In: *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, Las Vegas, USA. pp. IEEE: 4985–4988

[12] Do Q T N, Bethard S and Moens M-F 2015 Domain adaptation in semantic role labeling using a neural language model and linguistic resources. *IEEE/ACM Trans. Audio Speech Lang. Process.* 23

[13] Gildea D and Jurafsky D 2002 Automatic labeling of semantic roles. *Comput. Ling.* 28: 245–288

[14] Yan S and Wan X 2014 SRRank: leveraging semantic roles for extractive multi-document summarization. *IEEE/ACM Trans. Audio Speech Lang. Process.* 22

[15] Bhaskar S and Chandran S 2015 Semantic parsing approach in malayalam for machine translation. *Int. J. Eng. Res. Technol. (IJERT),* ISSN: 2278-0181 IJERTV4IS070698 www.ijert.org, Vol. 4

[16] Radhika K T and Reghu Raj P C 2013 Semantic role extraction and general concept understanding in malayalam using Paninian grammar. *Int. J. Eng. Res. Dev.* e-ISSN: 2278-067X, p-ISSN: 2278-800X, www.ijerd.com 9: 28–33

[17] Deshpande A R and Lobo L M R J 2013 Text summarization using clustering technique. *Int J. Eng. Trends Technol. (IJETT)* 4

[18] Sripada S, Kasturi V G and Parai G K 2005 Multi-document extraction based Summarization. CS 224N, Final Project, Stanford University, California, USA. pp. 1–8

[19] Wajid M S, Maurya S and Vaishya R 2013 Sentence similarity based text summarization using clusters. *Int J Sci Eng Res* 4, May-2013 1959 ISSN 2229-5518

[20] Kathiravan A V and Kalaiyarasi P 2015 Sentence-similarity based document clustering using birch algorithm. *Int. J. Innov. Res. Comput. Commun. Eng.* 3, ISSN(Online): 2320-9801 ISSN (Print): 2320-9798

[21] Zhang T, Ramakrishnan R and Livny M 1997 BIRCH: a new data clustering algorithm and its applications. *Data Min. Knowl. Discovery* 1: 141–182

[22] Patil M S, Bewoor M S and Patil S H 2014 Enhancing the performance of cluster based text summarization using support vector machine. *IJRET: Int. J. Res. Eng. Technol.* 3(12):53–58

[23] Ghorpade-Aher J and Metre V A 2014 Clustering multidimensional data with PSO based algorithm. In: *Third Post Graduate Symposium on Computer Engineering*, MCERC Nasik, India. pp. 53–56

[24] Baraldi A N and Enders C K 2009 An introduction to modern missing data analyses. *J. Sch. psychol.* 48(1):5–37

[25] Kohonen and Teuvo 2013 Essentials of the self-organizing map. *Neural Netw.* 37: 52–65

[26] Vesanto J and Alhoniemi E 2000 Clustering of the self-organizing map. *IEEE Trans. Neural Netw*. 11: 586–600

[27] Teuvo K 1990 The self-organizing Map. *Proc. IEEE* 78

[28] Ritter H, Martinetz T, Schulten K, Barsky D, Tesch M and Kates R 1992 *Neural computation and self-organizing maps: an introduction* (pp. 141–161). Reading, MA: Addison-Wesley

[29] Stefanovic P and Kurasova O 2011 Visual analysis of self-organizing maps. *Nonlinear Anal. Modell. Control* 16: 488–504

[30] Vesanto J and Alhoniemi E 2000 Clustering of the self-organizing map. *IEEE Trans. Neural Netw.* 11(3):586–600

[31] Blazejewski A and Coggins R 2006 *Application of self-organizing maps to clustering of high-frequency financial data.* School of Electrical and Information Engineering, University of Sydney, Sydney, Australia. Vol. 32, pp. 85–90

[32] Prajitha U, Sreejith C and Reghu Raj P C 2013 LALITHA: a light weight Malayalam stemmer using suffix stripping method. In: *2013 International Conference on Control Communication and Computing (ICCC). IEEE*

[33] https://olam.in/

[34] http://malayalamwordnet.cusat.ac.in/downloads.do

[35] Peng L and Lei L 2005 A review of missing data treatment methods. Intell. Inf. Mgmt. Syst Technol 1(3):412–419

[36] https://www.manoramaonline.com/home.html

[37] Indu M and Kavitha K V 2016 Review on text summarization evaluation methods. In: *International Conference on Research Advances in Integrated Navigation Systems (RAINS - 2016),* April 06–07, 2016, R. L. Jalappa Institute of Technology, Doddaballapur, Bangalore, India, 978-1-4673-8819-8/16/$31.00 ©2016 IEEE

[38] Saggion H and Lapalme G 2000 Concept identification and presentation in the context of technical text summarization. NAACL-ANLP 2000 Workshop: Automatic Summarization. university of montreal, Canada, pp. 1–10