Graph based Extractive Multi-document Summarizer for Malayalam-an Experiment

Manju K, David Peter S, and Sumam Mary Idicula

***Abstract*—Multidocument summarization is an automatic process to generate summary extract from multiple documents written about the same topic. Of the many summarization systems developed for English language, the graph based system is found to be more effective. This paper mainly focuses on a multidocument summarizing system for Malayalam Language which follows a graph based approach. 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. Experimental results demonstrate the effectiveness of the proposed system.**

***Index Terms*—Multi-document, summarization, extractive, graph, pagerank.**

I. Introduction

AUTOMATIC multidocument summarizer extracts information from multiple documents and provides it in a condensed form. In this digital age there is an inundation of information related to any topic. This necessitates the need for development of a system which can get information from multiple documents and provide it in a summarised form. Malayalam is an important regional language in India, predominantly spoken by the people of Kerala. The amount of digitised information available in Malayalam is also increasing rapidly with time. The source inputs for the summarization system can be news articles related to a specific topic from different Malayalam news paper dailies. This work helps to get important information from different Malayalam newspapers without redundancy. Automatic text summarization is an extremely active research field having connection with other research areas such as natural language processing, information retrieval and machine learning.

Two methods are widely followed in text summarization-Extractive and Abstractive. Extractive summarization system extracts the important sentences from the source documents without information loss while Abstractive summarization system generates summary by re-producing new sentences by taking the semantic meaning of sentences. Summary generation can be query relevant or generic. In query relevant system, summary generation will be based on the search term where as generic system provides an overall summary of the information contained in a document.

This work aims to develop an automatic summarization system that takes multiple Malayalam documents as input and generates a generic extractive summary as output. There are different methods to perform extractive summarization, of which graph based system gives a good result. Literature review of work conducted in this domain shows that no notable work has happened in Malayalam language for multidocument summarization. We can find some works in Hindi[9] and Bengali[10] related to single document summarization. The work presented in this paper intends to generate an extractive summary from multiple Malayalam documents, following a graph based ranking algorithm.

The rest of the paper is organized as follows: Section 2 reviews the previous work on document summarization that applies graph-based ranking algorithms. Section 3 discusses the methodology used, Section 4 discusses the experimental results and finally Section 5 concludes the paper.

Manuscript received March 06, 2016; revised March 31, 2016.

Manju K. is Research Scholar with the Department of Computer Science, Cochin University of Science and Technology, Kerala, India.(corresponding author e-mail: manju@mec.ac.in.)

David Peter S. is Professor with the Department of Computer Science and Engineering, Cochin University of Science and Technology, Kerala, India.

Sumam Mary Idicula is Professor with the Department of Computer Science, Cochin University of Science and Technology, Kerala, India.

II. Related Work

Graph based methods like LexRank[1] and TextRank[2] model document or set of documents as a text similarity graph constructed by taking sentences as vertices and the similarity between sentences as edge weight. They calculated the sentence significance from the entire graph using approaches which are inspired from Page Rank[7] and HITS[8] that were successfully applied to rank Web Page.

In work[3] a weighted bipartite graph was built on the document considering terms and sentences. An edge existed between terms and sentence if the term appeared in the sentence. This work applied the mutual reinforcement principle to find the saliency score of the sentences and the sentences were ranked in accordance with the saliency score.

The document-sensitive graph model[4] that emphasizes the influence of intra document relation on inter document relation. This model uses an extended form of page ranking algorithm to rank the sentences.

In work[5] the author treats extractive summarization by modelling documents by means of similarity graph and selecting sentences by Archetypal Analysis (AA).

III. Methodology

The proposed approach is a graph based multi-document extractive summarization method for Malayalam Language similar to LexPageRank. This system generates summary of a collection of articles taken from different newspapers on a specific topic. The process flow is as follows.

- Preprocessing: The plain text sources of the news articles require significant preprocessing because of the complexity associated with Malayalam Language.
- Graph Representation: Representing the documents as a graph in which each sentence will be a node and the edges will be the similarity between nodes.
- Sentence scoring: Scoring the sentence using Page Rank Algorithm.

ISBN: 978-988-19253-0-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

- Summary Generation. Sort the sentences on the score and generate summary in accordance with the compression ratio.

A. Preprocessing

1) Sentence Extraction: This module identifies sentence boundaries which can be a period(.), a question mark(?) or an exclamation mark(!). The period(.) symbol before recognizing as a sentence delimiter the module checks whether its part of abbreviation or part of a name or a decimal number or time. This is done by a set of rules represented using regular expression by considering the contexts where it appears. After extracting sentences from the documents, the stop words are removed.

2) Stopword Removal: Stop words are a set of commonly used words in any language. Removing stop words contribute to summarization scores. Most frequent words with no semantic content relative to the domain selected is the procedure for stop word selection. Here we have identified a list of stop words with respect to the data set considered for summarization. Sentences are scanned and the stop words are removed and the resulting sentences are obtained.

'അതെ', 'ഓട്', ഓടെ', 'ആയ', 'തുടങ്ങിയവർ', 'ഈ', 'ആണ്', 'ഇനി', 'ചില', 'പേർ', 'ഏറ്റവും', 'പുതിയ', 'എന്നാൽ', 'ഇന്നലെ', 'ആയി', 'എന്നിവരെ', 'എന്നീ', 'പല', 'അന്ന്', etc.

Fig. 1. Stop word list.

3) Stemming: This module converts a word into its root form. Even though literatures related to stemming in Malayalam language is available, there is no full fledged tool which can be used in our work. We have made some modifications on the Silpa Stemmer[11] which uses a suffix stripping algorithm[6]. The stemmer removes longest matching suffix from each word to get the base word.

For example the words *"vanathiloode"* and *"vanathil"* gets transformed to the root form *"vanam"*.

The stemming algorithm does this by by using a set of rules called stem rules. The stem rules are created by considering the different inflectional forms in Malayalam.

The stem rule used in the above example is *"thiloode"* and *"thil"* changes to *"m"*. The list of stem rules are given as stem rule file. While processing, the rule file is compiled and the values are moved to a dictionary like structure. The module also contains an exception pool where the exceptional words are directly mapped.

Example: The word *"makal"* is an exception, here *"kal"* is not a plural suffix.

Each sentence will be passed through the stemming module and the inflected words in them will be changed to its root form.

B. Document set as Graph

The document collection D is modeled as an undirected graph $G = (V, E)$ where V is the set of vertices and each vertex v_i in V is a preprocessed sentence of the document set. Here each vertex v_i in V will be represented in the form *doc.no._sent.pos* as *2_3* which means sentence 3 of document 2. E is the set of edges. Each edge e_i in E will be having a weight $f(v_i, v_j)$, showing the similarity between v_i and v_j $(i \neq j)$. The weight is calculated using the standard cosine measure between two sentences.

Cosine similarity between two sentences is the dot product of their vector representations. Here we have used the $Tf - Idf$ vector to represent the sentences.

$Tf - Idf$ Score: 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 determining the importance of a sentence. A Vector Space model is built at the sentence level by grouping all the sentences of the documents. Now for scoring the sentences, we determine the $Tf - Idf$ of each sentence in a document.

$$Tf - Idf(S_i) = Tf_{t,i} * Idf_t \tag{1}$$

where $Tf_{t,i}$ is the number of times the term t occurs in the sentence S_i and Idf_t gives the information about the number of sentences in which the term t appears.

$$Idf_t = log\frac{N}{N_t} \tag{2}$$

where N is the total sentences in a document D and N_t is the number of sentences in a document D in which the term t occurs. Taking the sum of $Tf - Idf$ of each term t in the sentence, we get the $Tf - Idf$ score of each sentence in the document. Since longer sentences will be having more no.of terms, we optimize the score by applying L2 Normalization.

Using the $Tf - Idf$ Vector we are computing the cosine measure between sentences.

$$f(v_i, v_j) = Sim(S_i, S_j) = \frac{S_i * S_j}{\sqrt{S_i^2} * \sqrt{S_j^2}} \tag{3}$$

If the similarity measure is larger than 0 then there is an edge between the vertices.

We use adjacency matrix A to describe G with each entry corresponding to the weight of an edge in the graph.

$$A_{i,j} = \left\{ \begin{array}{c} f(v_i, v_j),\ if v_i, v_j\ is\ connected \\ and\ i\ not\ equal\ to\ j \\ 0,\ otherwise \end{array} \right\} \tag{4}$$

A is normalized to A' where sum of each row is 1.

$$A'_{i,j} = \left\{ \begin{array}{c} A_{i,j} / \sum_{j=1}^{|v|} A_{i,j},\ if\ sum_{j=1}^{|v|} A_{i,j} \neq 0 \\ 0,\ otherwise \end{array} \right\} \tag{5}$$

C. Scoring Sentences

Based on the matrix A', the score of each sentence can be obtained by running the page ranking algorithm on the graph.

Page Rank[7] is a popular ranking algorithm developed by Google as a method for web link analysis. Even though traditionally Page Rank is applied on directed graph, this can also be applied to undirected graph, where the outdegree of the vertex is same as the indegree.

$$PR(V_i) = (1-d) + d * \sum_{V_j \varepsilon In(V_i)} w_{ji} \frac{PR(V_j)}{\sum_{V_k \varepsilon Out(V_j)} * w_{kj}} \tag{6}$$

ISBN: 978-988-19253-0-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

where d is the damping factor set between 0 and 1. The default value of d is 0.85 in Google[7].

The convergence of the algorithm is achieved when the difference in scores computed at successive iterations for any sentences fall below a threshold (here 0.0001). After page ranking algorithm is run on the graph, sentences are sorted in descending order of their scores. These are the candidate summary sentences.

D. Summary Generation

The summary sentences are selected from the candidate summary list. Depending on the compression ratio selected, top ranked sentences are included in the summary without any information overlap.

IV. Experimental Results and Discussion

As text summarization in Malayalam is a new field of research, no standard data set is available in this area. Therefore to test the accuracy of the multi-document summarization(MDS) system, five sets of document collection each having two or three related articles was taken from different news paper websites. The articles extracted from websites were saved as text files in UTF-8 format. For each document set a human generated summary was used for evaluation. An intrinsic evaluation scheme was employed by comparing the system generated summary with the human generated summary. If the compression ratio is 70%, the summary length will be 30% of the length of the largest file. The summarization system selects representative sentences from these input documents to form an extractive summary. The common information retrieval metrics, precision and recall are used to evaluate the new summary.

Precision is the fraction of system sentences that were correct.

$$Precision = \frac{system\ to\ human\ choice\ overlap}{sentences\ chosen\ by\ system}$$

Recall is the fraction of sentences chosen by the person, that were also correctly identified by the system.

$$Recall = \frac{|system\ to\ human\ choice\ overlap|}{|sentences\ chosen\ by\ human|}$$

F-measure is defined as the harmonic mean of Precision and Recall.

$$F - Score = \frac{2 * P * R}{P + R}$$

The system was tested and analysed on the data set selected at different compression ratio. On an average the system performed well for 70% compression ratio. The performance of the system was also affected by the document collection selected and varied with increase in the number of articles in each collection. As the compression ratio decreases there is a tendency of redundancy in the summary towards the end.

TABLE I
Performance analysis

Score	30% Compression	50% Compression	70% Compression
Precision	0.57	0.53	0.55
Recall	0.63	0.56	0.59
F-Score	0.8	0.66	0.72

A. Example

The system takes two related documents as input first document with eleven sentences and the second document with eight sentences. The system reads each sentence from the document and moves to a data structure where it stores the sentences along with an id which corresponds its document and the sentence position as in Fig.2.

1_1 ഡൽഹിയിൽ ബി.എസ്.എഫിന്റെ ചെറുവിമാനം തകർന്നുവീണ് 10 പേർ മരിച്ചു.
1_2 ബി.എസ്.എഫിന്റെ എഞ്ചിനീയറിങ് ടീമിലെ എട്ട് പേരും രണ്ട് പൈലറ്റുമാരുമാണ് വിമാനത്തിലുണ്ടായിരുന്നത്.
1_3 സെക്ടർ എട്ട് ദ്വാരകയിലെ ബഗ്ഡോള ഗ്രാമത്തിലാണ് വിമാനം തകർന്നുവീണത്.
1_4 ഇന്ദിരാ ഗാന്ധി അന്താരാഷ്ട്ര വിമാനത്താവളത്തിന് സമീപം ചൊവ്വാഴ്ച രാവിലെ 9: 50 ഓടെയായിരുന്നു സംഭവം.
1_5 9:45 നാണ് വിമാനം ടേക്ക് ഓഫ് ചെയ്തത്.
1_6 അഞ്ച് മിനിറ്റുള്ളിൽ തകർന്നുവീഴുകയും ചെയ്തു.
1_7 വിമാനത്താവളത്തിൽ നിന്ന് പറന്നയരവേ ഒരു മതിലിൽ തട്ടിയാണ് വിമാനം തകർന്നുവീണത്.
1_8 കനത്ത മൂടൽമഞ്ഞുണ്ടായിരുന്നു സംഭവസമയത്ത്.
1_9 പതിനഞ്ച് അഗ്നിശമന യൂണിറ്റുകൾ ചേർന്ന് തീയണച്ചു.
1_10 ബി.എസ്.എഫിന്റെ സൂപ്പർകിങ് എയർക്രാഫ്റ്റാണ് അപകടത്തിൽ പെട്ടത്.
1_11 റാഞ്ചിയിലേക്ക് പോകംവഴിയാണ് അപകടം സംഭവിച്ചത്.
2_1 പടിഞ്ഞാറൻ ഡൽഹിക്ക് സമീപം ദ്വാരകയിൽ ബിഎസ്എഫ് വിമാനം തകർന്നു വീണു പത്തുപേർ മരിച്ചു.
2_2 ഡെപ്യൂട്ടി കമാൻഡർമാർ ഉൾപ്പെടെ പത്തുപേരായിരുന്നു വിമാനത്തിൽ ഉണ്ടായിരുന്നത്.
2_3 പൈലറ്റുമാരായ രാജേഷ് ശിവറെയ്ൻ, ബി.പി. ഭട്ട്; ഇൻസ്പെക്ടർ ആർ.പി.യാദവ്; ഡപ്യൂട്ടി കമാൻഡന്റ്: ഡി.കുമാർ, ഇൻസ്പെക്ടർ: എസ്.എൻ . ശർമ്മ, എസ്ഐ: രവീന്ദ്ര കെ.ആർ, എസ്ഐ: ഛോട്ടേ ലാൽ, എഎസ്ഐ: ഡി.പി. ചൗഹാൻ , എസ് ഐ: സുന്ദർ സിങ്, കെ.റാവത്ത് എന്നിവരാണ് മരിച്ചത്.
2_4 ഇന്നു രാവിലെ 9.50 നാണ് സംഭവം.
2_5 ഡൽഹിയിൽ നിന്നും റാഞ്ചിയിലേക്ക് പുറപ്പെട്ട വിമാനമാണ് തകർന്നു വീണത്.
2_6 വിമാനം പറന്നയർന്ന് ഏതാനും മിനിറ്റുകൾക്കുള്ളിൽ സാങ്കേതിക തകരാറിനെ തുടർന്ന് തിരിച്ചിറക്കാൻ അനുവാദം ചോദിച്ചിരുന്നു.
2_7 ഇതിനു ശ്രമിക്കുന്നതിനിടെ സമീപത്തെ മതിലിൽ ഇടിച്ച് റയിൽവേ പാളത്തിന് സമീപം വീഴുകയായിരുന്നു.
2_8 ജനവാസമേഖലയിലാണ് അപകടമുണ്ടായത്.

Fig. 2. Input list holding the sentences and their id which shows the document in which it belongs and its position.

Now the sentences undergo preprocessing and the cosine similarity matrix is constructed based on which the adjacency matrix is constructed.The similarity matrix for the above example is as in Fig.3.

```
[[ 0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0. ]
 [ 0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0. ]
 [ 0.   0.   0.   0.   0.   0.   0.24 0.12 0.2  0.13 0.   0.15 0.1  0.26 0.   0.12 0.15 0.   0.   0. ]
 [ 0.   0.   0.   0.   0.23 0.   0.18 0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0. ]
 [ 0.   0.   0.   0.23 0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.16 0.18 0.   0.   0.   0. ]
 [ 0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.18 0.27 0.   0.   0.   0.   0.   0. ]
 [ 0.   0.   0.24 0.18 0.   0.   0.   0.13 0.   0.14 0.   0.16 0.   0.15 0.12 0.   0.17 0.   0.   0. ]
 [ 0.   0.   0.12 0.   0.   0.   0.13 0.   0.   0.13 0.   0.15 0.   0.14 0.   0.   0.15 0.   0.   0. ]
 [ 0.   0.   0.2  0.   0.   0.   0.   0.   0.   0.   0.   0.   0.19 0.   0.   0.   0.   0.   0.   0. ]
 [ 0.   0.   0.13 0.   0.   0.   0.14 0.13 0.   0.   0.   0.17 0.   0.29 0.   0.   0.17 0.   0.   0. ]
 [ 0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0. ]
 [ 0.   0.   0.15 0.   0.   0.18 0.16 0.15 0.   0.17 0.   0.   0.18 0.   0.   0.19 0.   0.   0.   0. ]
 [ 0.   0.   0.1  0.   0.   0.27 0.   0.   0.19 0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0. ]
 [ 0.   0.   0.26 0.   0.   0.   0.15 0.14 0.   0.29 0.   0.18 0.   0.   0.13 0.   0.18 0.   0.   0. ]
 [ 0.   0.   0.   0.   0.16 0.   0.12 0.   0.   0.   0.   0.   0.   0.   0.13 0.   0.26 0.   0.   0. ]
 [ 0.   0.   0.12 0.   0.18 0.   0.   0.   0.   0.   0.   0.   0.   0.   0.26 0.   0.   0.   0.   0. ]
 [ 0.   0.   0.15 0.   0.   0.   0.17 0.15 0.   0.17 0.   0.19 0.   0.18 0.   0.   0.   0.   0.   0. ]
 [ 0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0. ]
 [ 0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0. ]]
```

Fig. 3. Cosine similarity matrix.

From the similarity matrix the graph is constructed with sentences as nodes and their similarity as edges.The graph obtained is as in Fig:4.

Now page rank algorithm is applied on the graph.The values of the normalized adjacency matrix is used for the

ISBN: 978-988-19253-0-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

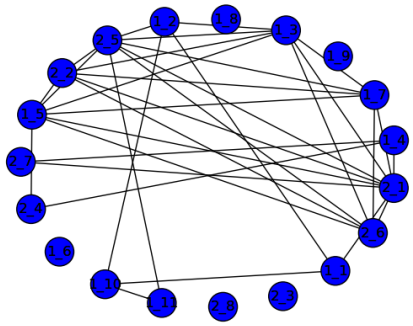

Fig. 4. Similarity Graph.

initial iteration. After convergence of the algorithm the page rank scores are sorted.The result is as in Fig:5. Each entry is a combination of sentence-id and page rank score. The sentence-id denotes the document and the position of the sentence in the document.

[(u'2_1', 0.11023205923958423), (u'2_5', 0.097510202823440556),
(u'1_3',0.095092082018057647), (u'1_5', 0.088242151114225773),
(u'1_7', 0.074606509611447852), (u'2_2', 0.073302691096338371),
(u'2_6', 0.061219594426259134), (u'1_2', 0.059807155587159738),
(u'1_10', 0.055914807719541548), (u'1_4', 0.055657533793612643),
(u'1_1', 0.052924677532629472), (u'2_4', 0.044880755717147246),
(u'1_11', 0.040881586697191816), (u'2_7', 0.038880734996245538),
(u'2_8', 0.010169491525423733), (u'2_3', 0.010169491525423733),
(u'1_6', 0.010169491525423733), (u'1_9', 0.010169491525423733),
(u'1_8', 0.010169491525423733)]

Fig. 5. Page Rank score of each sentence in sorted order.

With 50% compression ratio the summary generated from the system will have 6 sentences as in Fig:6.

പടിഞ്ഞാറൻ ഡൽഹിക്ക് സമീപം ദ്വാരകയിൽ ബിഎസ്എഫ് വിമാനം തകർന്നു വീണു പത്തുപേർ മരിച്ചു.(2_1) ഡൽഹിയിൽ നിന്നും റാഞ്ചിയിലേക്ക് പുറപ്പെട്ട വിമാനമാണ് തകർന്നു വീണത്.(2_5) സെക്ടർ എട്ട് ദ്വാരകയിലെ ബഗ്ഡോള ഗ്രാമത്തിലാണ് വിമാനം തകർന്നുവീണത്.(1_3) 9:45 നാണ് വിമാനം ടേക്ക് ഓഫ് ചെയ്തത്.(1_5) വിമാനത്താവളത്തിൽ നിന്ന് പറന്നുയരവേ ഒരു മതിലിൽ തട്ടിയാണ് വിമാനം തകർന്നുവീണത്.(1_7) ഡെപ്യൂട്ടി കമാൻഡർമാർ ഉൾപ്പെടെ പത്തുപേരായിരുന്നു വിമാനത്തിൽ ഉണ്ടായിരുന്നത്.(2_2)

Fig. 6. System generated summary.

The human generated summary in six sentences taken as the standard for evaluation is shown in Fig:7.

പടിഞ്ഞാറൻ ഡൽഹിക്ക് സമീപം ദ്വാരകയിൽ ബിഎസ്എഫ് വിമാനം തകർന്നു വീണു പത്തുപേർ മരിച്ചു. ഡൽഹിയിൽ നിന്നും റാഞ്ചിയിലേക്ക് പുറപ്പെട്ട വിമാനമാണ് തകർന്നു വീണത്. ഇന്ദിരാ ഗാന്ധി അന്താരാഷ്ട്ര വിമാനത്താവളത്തിന് സമീപം ചൊവ്വാഴ്ച രാവിലെ 9: 50 ഓടെയായിരുന്നു സംഭവം. വിമാനത്താവളത്തിൽ നിന്ന് പറന്നുയരവേ ഒരു മതിലിൽ തട്ടിയാണ് വിമാനം തകർന്നുവീണത്. ബി.എസ്.എഫിന്റെ സൂപ്പർകിങ് എയർക്രാഫ്റ്റാണ് അപകടത്തിൽ പെട്ടത്. ഡെപ്യൂട്ടി കമാൻഡർമാർ ഉൾപ്പെടെ പത്തുപേരായിരുന്നു വിമാനത്തിൽ ഉണ്ടായിരുന്നത്.

Fig. 7. Human generated summary.

V. Conclusion

An extractive multi-document system for Malayalam Language based on a graph representation for the text is developed. This paper shows that graph based methods for MDS of texts in Malayalam produces a relevant summary from multiple documents. It was observed that as the generated summary had sentences selected from different articles, there is a possibility of cross document co-reference. Since there is no existing system for multi-document Malayalam summarization, this serves as an introduction.

As graph based analysis does not require detailed linguistic knowledge, nor domain, it is highly portable to other domains and languages.

References

[1] Erkan, Günes and Radev, Dragomir R, *LexRank: Graph-based Lexical Centrality As Salience in Text Summarization*, 3rd ed. J. Artif. Int. Res., July 2004.
[2] Rada Mihalcea and Paul Tarau, *TextRank: Bringing order into texts IProceedings of EMNLP*, vol. 4, no. 4, pp. 404-411, 2004.
[3] Zha, Hongyuan, *Generic Summarization and Keyphrase Extraction Using Mutual Reinforcement Principle and Sentence Cluster‘ Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2002.
[4] Furu Wei Wenjie Li Qin Lu Yanxiang He, *A document-sensitive graph model for multi-document summarization*in *Knowledge and Information Systems* , 2010
[5] Ercan Canhasi , Igor Kononenko, *Multi-document summarization via Archetypal Analysis of the content-graph joint model* in *Knowledge and Information Systems.*
[6] Rajeev R.R., Rajendran N. and Elizabeth Sherly, *A suffix stripping based Morph Analyser for Malayalam language* in *Science Congress 2008.*
[7] Sergey Brin and Lawrence Page, *The Anatomy of a Large-Scale Hypertextual Web Search Engine* in *nfo-lab.stanford.edu/pub/papers/google.pdf*
[8] Jon Kleinberg, *Authoritative Sources in a Hyperlinked Environment* in*Proc. ACM-SIAM Symposium on Discrete Algorithms, 1998.*
[9] Tana, C.Malik, LateshThaokar, *Test model for summarizing hindi text using extraction method* in*Proc.2013 IEEE Conference on Information and Communication Technologies, ICT 2013.*
[10] Kamal Sarkar , *A Keyphrase-Based Approach to Text Summarization for English and Bengali Documents* in*International Journal of Technology Diffusion (IJTD), 5(2), 28-38. doi:10.4018/ijtd.2014040103.*
[11] *SILPA* “http://silpa.readthedocs.org/en/latest/index.html” *Copyright 2014, SILPA Project.*

ISBN: 978-988-19253-0-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)