International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 05 Issue: 07 | July 2018 www.irjet.net p-ISSN: 2395-0072

Extractive Text Summarization of Marathi News Articles

Yogeshwari V. Rathod

[1]Department of Computer Science & Engineering, Vishwakarma Institute of Technology, Pune

---

Abstract - *Text summarization is a process which defines summary as text which reflects the main and important sentences from the original text and preserving its information content and overall meaning. It is much more difficult task for human to create manually summary from large text document. We present technique for extractive summarization of news articles for Marathi language ,in which it will consists of selecting important sentences, paragraphs etc. from the original document and concatenating them into shorter form. In this paper, we develop system in two stages. First stage is Summarization of Domain Specific Marathi News. In Second stage we will extend our model for generic news will be tested on various Marathi news inputs. We can produce the summary of the article to varying degree of compression. Such a summarization technique is known for English articles, and doing it for Marathi news is the novel part of the work.*

Keywords: **Extractive text summarization, Text-rank algorithm , Page-rank algorithm.**

1. INTRODUCTION

Automatic text summarization is technique of compressing the original text into shorter form which will provide same meaning and information as provided by original text. The brief summary produced by summarization system allows readers to quickly and easily understand the content of original documents without having to read each individual document. Text summarization approaches can be classified into two groups: extractive summarization and abstractive summarization Extractive summaries involve extracting relevant sentences from the source text in proper order. The relevant sentences are extracted by applying statistical and language dependent features to the input text. [1]Abstractive summarization consists of understanding the source text by using linguistic method to interpret the text and expressing it in own language

In this paper, we introduce the Text-rank graph based ranking model for graphs extracted from natural language texts. We investigate and evaluate the application of Text rank to two language processing tasks consisting of unsupervised keyword and sentence extraction and show that the results obtained with text-rank. The text summarization software should produce the effective summary in less time and with least redundancy [2].

2. LITERATURE SURVEY

Several automatic text summarization systems are available for most of the commonly used natural languages. Maximum of these text summarization systems are for English and other foreign languages. Moreover, for commercial products the technical documentation is often minimal or even absent .For Indian languages, automatic text summarization systems are less .Various text summarizers for Indian languages are discussed below:

2.1 Bengali Language

Islam and Masum (2004) developed corpus oriented text summarization system for Bengali language. It is based on scoring the files of corpus in which query words are having highest frequency and then producing the summary of text documents on the basis of query words by applying vector-space-term-weighting[6].Bandyopadhyay(2010)developed Bengali opinion text summarizer based on given topic which can determine the information on sentiments in the input text. Then this information is aggregated for denoting text summary [9]. Sarkar(2012) proposed Bengali text summarization by sentence extraction and has investigated the impact of thematic term feature and position feature on Bengali text summarization. The proposed summarization method is extraction based [10].

2.2 Tamil language

Banu et al. (2007) proposed text summarizer for Tamil documents using technique of semantic graph by identifying Subject Object Predicate from individual lines for making semantic-graph of source text document and its corresponding summary generated by human experts[7]. Kumar and Devi (2011) proposed Tamil language summarization system for scoring of sentences in summary using graph theoretic scoring technique. This system uses statistics of frequency of words and a term

positional and weight-age calculation by string pattern for scoring of sentences [8].

2.3 Kannada Language

Kallimani et al. (2010) proposed a text summarizer for Kannada .This system processes the input text and then decides which lines are relevant and which lines are not relevant[11]. In it, text is summarized on console. Jayashree et al. (2011) proposed a text summarization system for Kannada named "Kannada text Summarizer based on Key terms Extraction"[12]. Jayashree et al. (2012) proposed another pre-classified documents summarizer for Kannada by scoring of sentences which retrieves key terms from Kannada documents, by combining GSS (Galavotti, Sebastiani, Simi) coefficients and Inverse-Document-Frequency techniques with Term Frequency for retrieving key term[13].

2.4 Punjabi Language

Gupta et al. (2012) proposed of Punjabi text summarizer. It makes extractive summary for Punjabi text by extracting the important lines based on language oriented features and features belonging to statistics of text [14].

2.5 Multi-Language

Keyan (2012) proposed multi-lingual (Tamil and English) multi-document summarization by neural networks. The proposed system can be able to summarize both Tamil and English online newspapers [15].

3. PROPOSED SUMMARIZATION METHOD

The proposed summarization method is extraction based. It has three major steps:

A. Pre-processing

B. Steaming

C. Sentence ranking for Summary generation.

A. Pre-processing

The pre-processing step includes stop-word removal, stemming and breaking the input document into a collection of sentences. For stop word removal, we have used the Marathi stop-word list downloadable from the website. From the given document of Marathi text, remove the punctuation mark characters like ; , ——: ()[]{} space character, tab space and so on for finding individual Marathi word. Marathi language has some stop words they frequently occurs words in Marathi text. We are Eliminate these words from text; otherwise, sentences containing them can get importance unnecessarily.

Table1: Marathi Stop Word Example

आहे	ते	कमी
या	असे	अनेक
आणि	होते	अधिक
व	केली	होणार
नाही	पण	म्हणाले
यानी	काही	याना
हे	केले	त्याची
तर	किवा	मी
आला	त्यामुळे	झाली
त	की	ता
येथे	न	टा

Pre-processing Module

This module deals with pre-processing of input text to and is divided into two parts filtration and tokenization. Figure 1 describes pre-processing module .

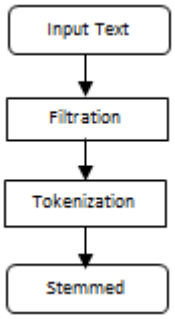

Figure 1: Pre-processing Module

In filtration input text is filtered out to remove any non-Devanagari Unicode but it is ensured that some punctuation marks like "_" and "-" are not excluded as they are also used in Marathi language word formation. Tokenization is the basic and important module of any NLP application.

Filtered Text: भारताची राजधानी नवी_ दिल्ली

b)Tokenized Text:

Token 0: भारताची

Token 1: राजधानी

Token 2: नवी दिल्ली

B. Stemming

Using stemming, a word is split into its stem and affix. The design of a stemmer is language specific, and requires some significant linguistic expertise in the language. A stemmer algorithm involves removing suffixes using a list of frequent suffixes. If word is found in Marathi noun or Marathi proper name list, its corresponding score is incremented by 1. If word is not found in Marathi noun or Marathi proper name list then Stemming is performed using any of stemming rules.

Stemmer contains two modules: pre-processing and stemming modules. Pre-processing output is provided as input to stemming module.

It is further subdivided into three parts:

(a) Root verification.

(b) Suffix removal.

(c) Inflection removal.

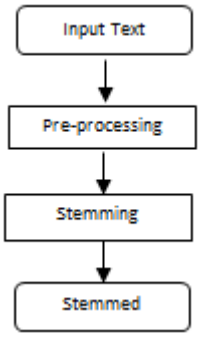

Figure 2: Stemming Module

Sample input and final output for Marathi stemmer.

Input Text: भारताची राजधानी नवी_ दिल्ली आहे.

Stemmed text: भारत राजधानी नवी_ दिल्ली

(a)Suffix removal

While splitting suffixes from base words it is first verified that length of a suffix is not larger than the length of the word to reduce chances of over and wrong stemming. To remove suffix we use predefined suffix list is created. shows some examples of suffix seen in the Marathi language.

Table2: Marathi Steam Word Example

Suffix	Marathi Word	Root Word
ला	रामला	राम
च	रामच	राम
चा	रामचा	राम
ची	रामची	राम
चे	रामचे	राम
च्या	रामच्या	राम
साठी	पुजाऱ्यासाठी	पुजारी
सोबत	पुजाऱ्यसोबत	पुजारी
सारखी	मुलीसारखी	मुलगी
सारखा	मुलासारखा	मुलगा
सारखे	मुलीसारखे	मुलगी
मुळे	गीतामुळे	गीता
बाबद	मुलाबाबद	मुलगा
बरोबर	मुलाबरोबर	मुलगा

(b) Inflection removal

Most of the Marathi words are inflected. Some inflection removal rules are as follows: If word after removal of suffix contains डा, डी, डे then convert it to ड. If word after removal of suffix contains ता, तत, ती, तु, ते then convert it to त for example in word भारताची after removing suffix ची word becomes भारता and here after applying inflection removal rules i.e. ता is converted into त and we get root word भारत.

(c) Sentence ranking

After an input document is formatted and stemmed, the document is divided into a collection of sentences and the sentences are ranked based on two important features: thematic term and position.

3. PAGERANK IMPLEMENTATION FOR MARATHI LANGUAGE

The graph-based ranking algorithm used to find the importance for the nodes i.e. vertex within the graph. When one vertex is connected to another one, it is basically casting recommendation for that other vertex. The higher the number of votes that are cast for a vertex, the higher the importance of the vertex[4].

Formally, let G= {V,E}be a directed graph with the set of vertices V and set of edges E, Where E is subset of V*V. For a given vertex Vi, Let In(Vi) be the set of vertices that point

to it (predecessors), and let Out(Vi) be the set of vertices that vertex Vi points to (successors). The score of a vertex Vi is defined as follows (Brin and Page, 1998):

$$S(Vi) = (1 - d) + d * \sum_{j \epsilon In(Vi)} \frac{1}{|Out(Vj)|} S(Vj)$$

Where

'V'= set of vertices,

'E'=set of edges,

V(in) = Set of incoming edges,

V(out) = Set of outgoing edges,

d = damping factor (default =0.85),

W = set of edge weights

For undirected graphs, V(in) = V(out)

To take care of a case when there is no linking for any node . The probability, at any step, that the person will continue is a damping factor d. It is usually set to .85

To rank Marathi texts, we have to create a graph having the Marathi word, and interconnect words or other text entities with meaningful relations. In graph text units of various sizes and characteristics can be included as vertices in the graph, e.g. words, collocations, entire sentences, or others [4].

Graph algorithms to for Marathi of the following main steps:

1. Identify text units that best define the task and add them as vertices in the graph.

2. Identify relations that connect such text units, and use these relations to draw edges between vertices in the graph. Edges can be directed or undirected, weighted or unweight.

3. Iterate the graph-based ranking algorithm until convergence.

4. Sort vertices based on their final score. Use the values attached to each vertex for ranking/selection decisions.

In the following, we investigate and evaluate the application of Text-rank to two natural language processing tasks involving ranking of text units:

(1) A keyword extraction task, consisting of the selection of keyphrases representative for a given text;

(2) A sentence extraction task, consisting of the identification of the most "important" sentences in a text, which can be used to build extractive summaries.

Pseudocode

OPEN and READ file

Filter ASCII(text) //Filter non ASCII characters

sentences = regex.split('(?<!\w\.\w.)(?<![A-Z][az]\.)(?<=\.|\?)\s', text)

FOR j in 0,length(sentences) //Split sentences

APPEND ([i for i in sentences[j].split() if i not in stop])

PunctuationRemove(sentences)

MarathiStemmer(sentences) //Apply Stemmer

ComputeFrequency(word in sentences)

fori in range(0,len(sentences)): //Compute Adjacency

for j in range(i+1,len(sentences)):

adjacency[i,j] = adjacency[j,i] = find_distortion(i,j)

adjacency=1-adjacency/float(adjacency.max())

fori in range(0,10): // 10 iterations for convergence

text_rank = find_rank(text_rank,adjacency,0.85)

print sentences.

3.1 Keypharse Extraction

This method is used to automatically identify text as a set of words which best describe the document. Such keywords can be useful entries for building an automatic index for a document collection, it is useful to classify a text to generate the summary. To apply Text-rank, we first need to build a graph associated with the text, where the graph vertices are representative of the units to be ranked. After sentence extraction, to rank all sentences, all vertex is added to the graph for each sentence from the original

text. The co-occurrence relation used for keyword extraction is used here. We will create a graph for keyphrase extraction using important keywords in our all sentences, we will rank sentences using those keywords generate the summary.

For Example :

1.नवीदिल्ली:भीषणपाणीटंचाईतअडकलेल्यालातूरकरांच्यामदतीसाठीआतादेशभरातअनेकमदतीचेहातयेण्याससुरुवातझालीआहे.
2.दिल्लीचेमुख्यमंत्रीअरविंदकेजरीवालयांनीदेखीलतहानलेल्यालातूर लापाणीदेण्याचीतयारीदर्शवलीआहे.
3. यासाठीत्यांनीपंतप्रधानमोदींनाएकपत्रलिहिलंआहे.

4. दिल्लीकरनागरिकदररोजपाणीवाचवून 10 लाखलिटरपाणीलातूरलादेण्यासतयारआहेत, फक्त केंद्रसरकार ने ते पाणी लातूर पर्यंत पोहचवण्याची व्यवस्थाकरावी असंयापत्रात म्हटलंआहे.
5. मिरजहून पाणी एक्स्प्रेस लातुरात दरम्यान, दुष्काळा नंत्रस्तझालेल्यालातुरकरांसाठीआजचीसकाळहीनव्यासुर्योदयासहएक नवीउमेदघेऊनआली.
6.कारण, गेलेकित्येकदिवसपाण्याचीआतुरतेनेवाटपाहणाऱ्यालातुर करांचीपाण्याचीप्रतीक्षाअखेरआजसंपुष्टातआली.
7. पाण्याचीमिरजएक्सप्रेसआजसकाळीलातुरातदाखलझाली. पाण्याचे 10 वॅगन्सयाएक्सप्रेसलाजोडण्यातआलेहोते.
8. एकावॅगनमधूनसाधारणपणे 50 हजारलिटरपाणीनेण्यातआलंआहे.
9.
त्यामुळेयापहिल्याखेपेलालातूरकरांनातब्बल५लाखलीटरपाणीमिळालं .
10.तसेचलवकरचउर्वरितवॅगनहीलवकरचलातूरकडेपाठवलीजाणार आहे.
11.दरम्यान, लातूरातभीषणपाणीटंचाईअसल्यानेसरकारयाआधीउ जनीधरणातूनरेल्वेनंलातूरलापाणीदेणारहोतं.
12. मात्रत्यातहीअडचणीआल्यानेमिरजचीनिवडकरण्यातआली. एखाद्याजिल्ह्यालारेल्वेनेपाणीदेण्याचीवेळमहाराष्ट्रावरपहिल्यांदाचओ ढावली आहे.

From above paragraph we will get important keywords like :

['पाणी', 'देण्', 'मिरज', 'पाण्', 'नवी', 'भीषण', 'लातूर', 'पाणीटंचाई', 'दिल्ली:', 'सरकार', '10', 'लवकरच', 'पहिल्यांदाच', 'लातुरा']

With the help these keyword we will rank the sentences and generate summary.

To avoid growth of the graph size which is more than necessary, by adding all possible combinations of sequences consisting of more than one lexical unit (n-grams), we have nominated only single words for addition to the graph, with multi-word keywords being finally reconstructed in the post-processing phase. Later on , all lexical units that pass the syntactic filter for Marathi language are added to the graph, and an edge is added between those lexical units that co-occur within a window of Marathi words. After the graph is constructed (undirected un-weighted graph), the score associated with each vertex which is Marathi word is set to an initial value of 1, and the ranking algorithm described in is run on the graph for several iterations until it converges – usually for 20-30 iterations, at a threshold of 0.0001.Once a final score is obtained for each vertex in the graph, vertices are sorted in reversed order of their score, and the top vertices in the ranking are retained for post-processing.

3.2 Text-rank for Sentence Extraction

This extraction technique is known for English language , We have implemented it for Marathi language .To use Text-rank algorithm for Marathi language, First we have to create a graph associated with the text , where the graph vertices are used to ranked the units. For sentence extraction in Marathi, the goal is to rank entire sentences, and therefore a vertex is added to the graph for each sentence in the text. The co-occurrence which we have used earlier for keyword extraction.

Cannot be applied here, since the text units in consideration are significantly larger than one or few words, and “co-occurrence” is not a meaningful relation for such large contexts, it can give some meaningless data if text structure is not valid. Instead, we are defining a different relation, which determines a connection between two sentences if there is a “similarity” relation between them, where “similarity” is measured as a function of their content overlap. This relation between sentences consider as process of endorsement”: a sentence that addresses certain concepts in a text, gives the reader a “recommendation” to refer to other sentences in the text that address the same concepts, and therefore a link can be drawn between any two such sentences that share common content.

The similarity of two sentences can be measured by the number of common tokens between two sentences, or it can be run through syntactic filters, which only count words of a certain syntactic category, e.g. all open class words stopwords etc. To avoid upholding long sentences, we are using a normalization factor, and break the content overlap.

For Example:

1.नवीदिल्ली:भीषणपाणीटंचाईतअडकलेल्यालातूरकरांच्यामदतीसा ठीआतादेशभरातअनेकमदतीचेहातयेण्याससुरुवातझालीआहे.
2.दिल्लीचेमुख्यमंत्रीअरविंदकेजरीवालयांनीदेखीलतहानलेल्यालातूर लापाणीदेण्याचीतयारीदर्शवलीआहे.
3. यासाठीत्यांनीपंतप्रधानमोदींनाएकपत्रलिहिलंआहे.
4. दिल्लीकरनागरिकदररोजपाणीवाचवून 10 लाखलिटरपाणीलातूरलादेण्यासतयारआहेत, फक्तकेंद्रसरकारनेतेपाणीलातूरपर्यंतपोहचवण्याचीव्यवस्थाकरावीअ संयापत्रातम्हटलंआहे.
5. मिरजहून पाणी एक्स्प्रेस लातुरात दरम्यान, दुष्काळानं त्रस्तझालेल्याला तुरकरांसाठीआजची सकाळही नव्यासुर्योदया सह एक नवी उमेद घेऊनआली.
6.कारण, गेलेकित्येकदिवसपाण्याचीआतुरतेनेवाटपाहणाऱ्यालातुर करांचीपाण्याचीप्रतीक्षाअखेरआजसंपुष्टातआली.
7. पाण्याची मिरज एक्सप्रेस आज सकाळी लातुरात दाखल झाली.

पाण्याचे 10 वॅगन्सयाएक्सप्रेसलाजोडण्यातआलेहोते.
8. एकावॅगनमधूनसाधारणपणे 50 हजारलिटरपाणीनेण्यातआलंआहे.
9.त्यामुळेयापहिल्याखेपेलालातूरकरांनातब्बल५लाखलीटरपाणीमि ळालं.
10.तसेचलवकरचउर्वरितवॅगनहीलवकरचलातूरकडेपाठवलीजाणार आहेत.
11.दरम्यान,लातूरातभीषणपाणीटंचाईअसल्यानेसरकारयाआधीउ जनीधरणातूनरेल्वेनंलातूरलापाणीदेणारहोतं.
12. मात्रत्यातही अडचणी आल्याने मिरजची निवड करण्यातआली. एखाद्या जिल्ह्याला रेल्वेने पाणी देण्याची वेळ महाराष्ट्रावर पहिल्यांदाचओढावलीआहे.

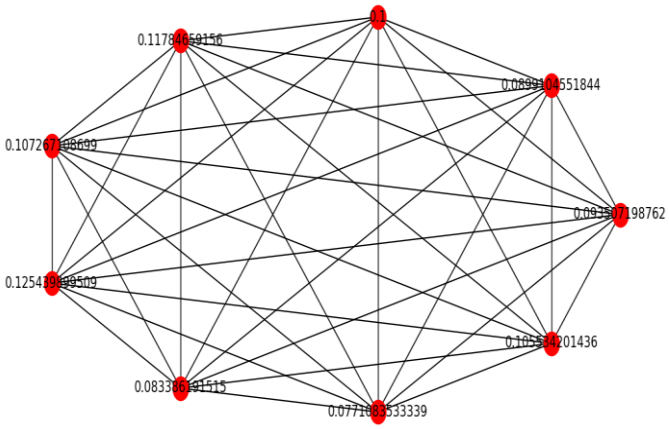

Figure: Sample graph build for sentence extraction from a newspaper article

Formally , given two sentences Si and Sj , with sentence being represented by the set of Ni words that appear in the sentence: Si = W1,W2,W3.....W Ni, the similarity of Si and Sj is

Defined as:

$$\boldsymbol{Similarity}(\boldsymbol{S_i}, \boldsymbol{S_j}) = \frac{|\{\boldsymbol{w_k}|\boldsymbol{w_k} \in \boldsymbol{S_i} \& \boldsymbol{w_k} \in \boldsymbol{S_j}\}|}{\log(|\boldsymbol{S_i}|) + \log(|\boldsymbol{S_j}|)}$$

Other sentence comparison measures, such as string, cosine similarity, longest common subsequence, etc. are also possible, and we are currently evaluating their impact on the summarization performance. The generated graph is extremely connected with each edge, with a weight associated indicating the strength of the connections established between various sentence pairs in the text. The text is therefore as a weighted graph, and consequently we are using the weighted graph-based ranking formula.

4. EVALUATION

It is found that there are mainly two categories of evaluation techniques: Intrinsic and Extrinsic . The Intrinsic methods use Human generated summaries for comparisons as they are considered to be intellectual summaries.. Intrinsic method has two approaches: content-based and co selection-based.. The sentence extracts, is often measured by co-selection. It counts how many reference summary sentences the candidate summary contains i.e. summary generated by automatic summarizer. The Content-based measures actually compare the words in a sentence, rather than the entire sentence. The main advantage of this measure is, it can be used to compare extractive summaries with abstractive summaries.

It measures the performance of summarization system by comparing a candidate summary with human generated summary known as reference summary or ideal summary.

We have created 2 summaries by human and compare with the automatic generated summaries using rough tool.

Table 3: Rough Evaluation

Summarizer	First File	Second File	Average
Similarity Based	0.842105263158	0.569620253165	0.705862758161
Position Based	0.842105263158	0.405063291139	0.623584277149

File : 1 Score : 0.842105263158

File : 2 Score : 0.569620253165

Average ROUGE-2 score 0.705862758161

From above result we can say that similarity based summarization Technique is more efficient and accurate.

5. CONCLUSION

In this paper, we have extended existing Text-rank algorithm for Marathi news articles graph-based ranking model for text summarization processing, and show how this model can be successfully used in natural language applications. In particular, we propose two innovative unsupervised methods for keyword and sentence extraction. Lots of work has been done in Bengali, Punjabi, Tamil, etc. languages. We have implemented it to generate summary for Marathi NEWS articles.

6. REFERENCES

[1]Virat V. Giri, Dr.M.M. Math and Dr.U.P. Kulkarni ,"A Survey of Automatic Text Summarization System for Different Regional Language in India"(2016).

[2]Vishal Gupta and Gurpreet Singh Lehal,"A Survey of Text Summarization Extractive Techniques"(Aug 2010).

[3] Gupta and G.S. Lehal, —Complete Preprocessing Phase of Punjabi Language Text Summarization‖, International Conference on Computational Linguistics COLING'12, IIT Bombay, India, Pp. 199205, 2012.

[4]Rada Mihalcea and Paul Tarau, "TextRank: Bringing Order into Texts"(July 2004).

[5] Frank, G. W. Paynter, I. H. Witten, C. Gutwin, and C. G. Nevill-Manning," Domain-specific keyphrase extraction".

[6]T Eslam and S.M.A. Masum ,"Bhasha: A Corpus Based Information Retrieval and Summarizer for Bengali Text"(2004).

[7] M. Banu, C. Karthika, P. Sudarmani and T.V. Geetha, —Tamil Document Summarization Using Semantic Graph Method‖, Proceedings of International Conference on Computational Intelligence and Multimedia Applications, Vol. 2, Pp. 128-134, 2007.

[8] S. Kumar, V.S. Ram and S.L. Devi, —Text Extraction for an Agglutinative Language‖, Proceedings of Journal: Language in India, Pp. 56-59, 2011.

[9] A. Das and S. Bandyopadhyay, —Topic-based Bengali opinion summarization‖, In Proceedings of the 23rd International Conference on Computational Linguistics: Posters, Pp. 232–240, 2010.

[10] K. Sarkar, —An approach to summarizing Bengali news documents‖. In proceedings of the International Conference on Advances in Computing, Communications and Informatics, Pp. 857-862, 2012.

[11]J.S. Kallimani, K.G. Srinivasa and B. R. Eswara,—Information Retrieval by Text Summarization for an Indian Regional Language,‖ In Proceedings ofInternational Conference on Natural Language Processing and Knowledge Engineering, pp. 1-4, 2010.

[12] R. Jayashree, K.M. Srikanta and K. Sunny, —Document Summarization in Kannada using Keyword Extraction‖, Proceedings of AIAA 2011,CS& IT 03, Pp. 121–127, 2011.

[13] R. Jayashree, —Categorized Text Document Summarization in the Kannada Language by Sentence Ranking‖, Proceedings of 12th International Conference on Intelligent Systems Design and Applications (ISDA), Pp. 776-781, 2012.

[14] Gupta and G.S. Lehal, —Complete Preprocessing Phase of Punjabi Language Text Summarization‖, International Conference on Computational Linguistics COLING'12, IIT Bombay, India, Pp. 199205, 2012.

[15] M.K. Keyan and K.G. Srinivasagan, —Multi-Document and MultiLingual Summarization using Neural Networks‖, Proceedings of International Conference on Recent Trends in Computational Methods, Communication and Controls,Pp. 11-14, Vol. 5, 2012.

[16]http://www.india.com/marathi/india/girish-mahajan-on-arvind-kejriwal/