International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 3 | Issue 3 | ISSN : 2456-3307

Extractive Summarizer Construction Techniques : A Survey

Vaishali V. Sarwadnya*1, Sheetal. S. Sonawane2

*1Pune Institute of Computer Technology, Pune, Maharashtra, India

2Pune Institute of Computer Technology, Pune, Maharashtra, India

ABSTRACT

Manual summarization of large documents of texts is tedious and error prone. Also, the results in such kind of summarization may lead to different results for a particular document. Thus, Automatic text summarization has become important due to the tremendous growth of information and data. It chooses the most informative part of text and forms summaries that reveal the main purpose of the given document. It yields summary produced by summarization system which allows readers to comprehend the content of document instead for reading each and every individual document. So, the overall intention of Text Summarizer is to provide the meaning of text in less words and sentences. Summarization can be categorized as: Abstractive summarization and Extractive summarization. This case study is based on an extractive concept implemented on the studied models. Numerous automatic text summarization systems are handy today for English and other foreign languages. But when it comes to Indian languages, we observe inadequate number of automatic summarizers. Evaluation can be done using quantitative or qualitative approach. This paper describes review of techniques used while constructing extractive summarizers and an approach to construct extractive summarizer for Marathi.

Keywords: Extractive Summarizer, Feature Extraction, Sentence Scoring, Marathi

I. INTRODUCTION

Text Summarization is a technique of condensing actual text into abstract form which provides same meaning and information as provided by actual text. It chooses the most informative part of text and forms summaries that reveal the main purpose of the given document. It yields summary produced by summarization system which allows readers to comprehend the content of document instead for reading each and every individual document. So, the overall intention of text summarizer is to provide the meaning of text in less words and sentences. Summarization systems can be sorted into two categories: Abstraction-based summarization and Extraction-based summarization.

Extractive summaries involve extracting appropriate sentences from the source text in sequential manner. The appropriate sentences are extracted by applying statistical and language reliable features to the input text. But there is limit in extraction. The extracted phrases and sentences are in chronological order. While, abstractive text summaries are formed by enacting natural language understanding concepts.

This kind of summarizer generally, incorporates terms that do not exist in the document. It aims to imitate methods used by humans, such as representing a concept that is available in the original article in a better and more comprehensive way. It is effective summarizer however, it is very difficult to implement.

CSEIT183152 | Received : 20 April 2018 | Accepted : 30 April 2018 | March-April-2018 [(3) 3 : 2058-2066]

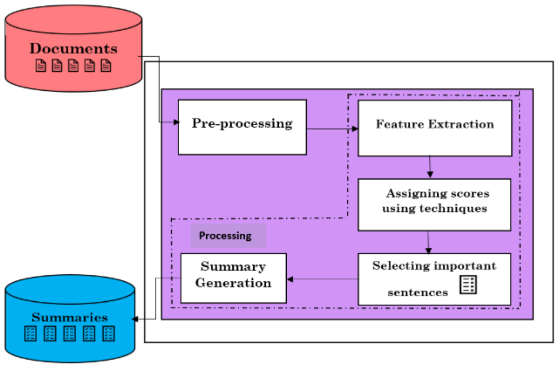

Figure 1. Extractive Text Summarizer System

A. Abstractive Summarization

In this type of summarization, language understanding tools are used to generate a summary. The main focus is on choosing phrases and lexical chains from the documents. General steps used in this technique are withdrawing basic features, obtaining the relevant information, revising and reducing information. Since the formulation of this technique in mathematical or logical form is cumbersome, it is referred as a complex technique to implement. Also, the quality of generated summaries relies on the depth of linguistic strength. These techniques are generally categorized as Structured and Semantic based. Structured based approaches, obtain most significant information from the documents through cognitive schemas such as templates, frames and scripts [3]. Semantic based approach makes use semantic depiction of documents which is further used to supply into natural language generation (NLG) system as input. This method focuses on obtaining noun phrases and verb phrases by managing linguistic data. Phrases thus obtained are then related to concepts, attributes and relations of a domain-specific ontology. The important document areas (like sentences or paragraphs) are selected by using ontology-based annotations and clustering techniques. The information obtained as an outcome is used to transform those areas into semantic representation. This NLG system takes this representation as input and then produces abstracts. Word parsing, rhetorical parsing, statistical parsing and a mixture of all are some of the common techniques used. The drawbacks of abstractive summarization approach are 1) Machine generated automatic summaries would result into lack of clarity even within a sentence as sentence synthesis is an emerging field.

2) As they heavily depend on the adaptation of internal tools to perform information extraction and language generation, they are difficult to replicate [3]. 3) Abstracting is not easier as it needs semantic understanding of content present in the document.

B. Extractive Summarization

Extractive summaries are formed by extracting crucial text (sentences or passages) from the document, based on features such as word/phrase frequency, location or cue words to locate the sentences to be extracted. The most important text is treated as the most frequent or the most suitably positioned text. Such an approach thus, shuns the labour on depth of content understanding. They are theoretically simple and easy to implement. This system constitutes two important phases, which are : Pre-Processing and Processing phase 1) Representing the text in a structured manner is the main aim of Pre-Processing phase . 2) Processing phase represents various features that decide the importance of sentences. Certain statistical features used for marathi language are keywords identification, sentence length feature and numerical literals count feature. An equation of summation of feature weights is used to generate score of sentences and high scored sentences in a specific order of input text are considered for final summary. This report describes multi document Marathi extractive summarizer. It is text extraction based summarization system which is used to summarize the Marathi document by retaining the appropriate sentences based on features.

II. RELATED WORK

Various automatic text summarization systems are commercially or noncommercially available for most of the commonly used natural languages. Most 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. When it comes to Indian languages, automatic text summarization systems are still lacking. Very little research and work has been done in text summarization for the Indian language Marathi (an Under-Resourced language). Text summarization is the process of extracting important information from the source text and to present that information to the user in the shorter version. In various fields text summarization is used like education field, social media (news articles, twitter, Facebook messages), biomedical field, government offices, researcher, etc [2]. Virat V. Giri, and et al. reviewed text summarizers based on various Indian languages and their performances. They studied and proposed summarization method for Marathi in detail wherein Marathi stemmer, Marathi proper name list, English-Marathi noun list, Marathi keywords extraction, Marathi rule based named entity recognition etc. for pre-processing of text followed by processing of text [1]. Sheetal Shimpikar and et al. studied various techniques of text summarization for various Indian languages [2]. Sunitha C and et al. worked on Abstractive summarization techniques in Indian languages. They explained Abstractive summarization technique, classified in two approaches such as structure based approach and semantic based approach[3]. Hamzah Noori Fejer and et al. gave a major contribution by proposing a combined approach of clustering method and keyphrase extraction. The is a new approach of clustering which combines between hierarchical clustering methods and k-means clustering. The results of their experiments proved the proposed model gives better performance in comparison to other works [4]. An unsupervised approach to Marathi stemmer has been discussed by Mudassar M. Majgaonker and et al. [5]. The present work on text summarization of Marathi text with question based system using rule based stemmer technique. For generating question, we used rule based approach of abstractive text summarization and POS tagger, NER tools and rule based stemmer. Here, Marathi text is taken as input, on it POS tagger is applied and then questions are generated for the given input as per Marathi language rules by Deepali K. Gaikwad and et al. At this stage they have framed rules of stemmer only for Who type questions [6]. Thus it can be extended to learning all What type questions too.

S. A. Babar and et al. proposed method that improves the quality of summary by incorporating the latent semantic analysis into the sentence feature extracted fuzzy logic system to extract the semantic relations between concepts in the original text[7]. Mangesh Dahale proposed text summarizer using inverted indexes [9]. Jayshri Patil and et al. reviewed different approaches of Named Entity Recognition (NER) and discussed issues and challenges arising in Indian languages [8]. Pooja Pandey and et al. discussed extraction of root words using morphological analyzer for devanagari script [15]. Aishwarya Sahani and et al. contributed to automatic text categorization of marathi language documents [7]. Rafael Ferriera et. al used four dimensional graph based model for text summarization which relies on four dimensions (similarity, semantic similarity, co-reference, discourse information) to create the graph [25]. Federico Barrios et. al used variations in similarity measures along with TextRank for summarization [19].

III. CHALLENGES OBSERVED FOR MARATHI LANGUAGE

Automatic text summarization has become important due to the rapid growth of information texts since it is very difficult for human beings to manually summarize large documents of texts. A full understanding of the document is essential to form an ideal summary. However, achieving full understanding is either difficult or impossible for computers. Therefore, selecting important sentences from the original text and presenting these sentences as a summary present the most common techniques in automated text summarization [4].

Various automatic text summarization systems are commercially or noncommercially available for most of the commonly used natural languages. Most 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. When it comes to Indian languages, automatic text summarization systems are still lacking [5]. Also very little work has been done for constructing a text summarizer for Marathi language. Marathi language is morphologically very rich. A single root word may have many different morphological variants like for word 'desh' we have deshachi, deshani, deshasathi, deshakarita, etc. morphological variants. Hence, it is required to study its morphology and pre-process the document before extracting features and then process those features which are important in each of the sentences.

There are certain challenges that need to faced while constructing an automatic text summarizer for Marathi language:

1. Marathi text is inflectional and morphologically rich [5]. Marathi is agglutinative language. Unlike English prefixes and suffixes are added to root words in Marathi to form meaningful contexts. Sometimes in Marathi inclusion of suffixes or prefix to root word leads to change in semantic. So a difficult and critical situation is raised to use gazetteers, dictionaries, similarity measurement and pattern matching techniques to recognize Marathi names. Dictionaries or gazetteers contain entities without any suffix added. In Marathi suffixes are added to words in order to create the meaningful context. A well written stemmer is required for morphologically rich language Marathi to separate the root from the suffix in order to compare the word forms with gazetteer or dictionary entries. Next, it cannot be claimed that stemming will solve the problem completely because adding suffixes to roots may change the grammatical category of the root word, which may result in wrong entity recognition [9].
2. Non-availability of large Gazetteer Its difficult to construct gazetteer for such a influential and morphologically variant language.
3. It is not easy, because Marathi language do not have capitalization.
4. Scarcity of resources
5. Ambiguities in names where the words have multiple interpretations while analyzing the text containing words.
6. In Marathi, words containing some vowels do not make phonetic difference but differs in writing and spellings [9].
7. Foreign words in some instances of person, organization, location and miscellaneous names that are English words appear in Marathi texts which are spelled in Devanagari script. The real challenge lies in recognition of such foreign words. It is very difficult to create gazetteers that include such names because they are not limited 8. Marathi is spoken using many dialects such as standard Marathi, Warhadi, Ahirani, Dangi, Vadvali, Samavedi, Khandeshi, and Malwani in various regions of India. There are specific words used in each dialect to express the text. Words from different dialects also appear in Marathi text.

IV. PRE-PROCESSING STAGE

Pre-processing stage is essential in text summarization. It results into pre-processed data, which is ideally fit for processing stage. We provide Marathi language document(s) to the text summarizer which are relevant to specific topics like news, sports, etc. These are passed to pre-processing stage. In general pre-processing stage consists of steps to remove punctuation marks, tokenization, stop word removal, stemming, etc. In this section we will discuss various steps used in pre-processing stage.

A. Boundary Identification and Punctuation marks removal

Every sentence ends with a punctuation mark depending on the nature of sentence, whether interrogative, exclamatory, imperative or declarative. Also, use of quotation marks ("",'), commas (,), special characters(&,*,|) and symbols(#,@), etc. is frequent. But when it comes to extract important words for processing stage, we need to eliminate these punctuation marks. Hence, we use techniques for removal of punctuation marks. The output of this step is punctuation marks free sentences in the document.

B. Stop words Elimination

Frequently occurring non-essential words for processing in text summarization are generally termed as stop words. In marathi language, we use stop words like shivay, ase, eetar etc. in day to day use. We should eliminate them for obtaining meaningful context while processing the documents. The output of this sentence is stop words free sentences in the document.

C. Stemming and lemmatization

The process of stemming is to obtain stem or radix of those words which aren't available in dictionary or morph. Lemmatization identifies lemma of a word. It maps the verbs into their infinitive and nouns into their singular form. Methods used for constructing stemmers include: Rule based Porter's Stemmer, Husk stemmer, Unsupervised stemming, suffix stripping Lovins stemmer, Dawson stemmer, Ngram method, HMM method, YASS (Yet Another Suffix Stripper) stemmer etc.

V. PROCESSING STAGE

Feature Extraction and Sentence Ranking Process The features like SOV (Subject Object Verb - Experimental) verification, sentence positional value (POS tagging), TF-ISF (Term Frequency/ Inverse Sentence Frequency) or TF-IDF (Term Frequency/ Inverse Document Frequency) are extracted from pre-processed sentences. Sentences are further ranked on basis of features extracted. Following are the general approaches followed:

1. Word Scoring - assigning scores to important words.
2. Sentence Scoring-sentence position, title name matching, etc.
3. Graph scoring - analyzing relationship between sentences.

A. Word Scoring

It observes features like Word frequency, TF/IDF, Proper noun, Word co-occurrence and Lexical similarity to determine respective scores of the words in the sentences. Each word obtains a score and by summing up the scores we obtain weights of each sentences.

i. Word Frequency: The assumption is that the higher the frequency of a word in the text, the more likely that it indicates the subject of the text [3].

ii. TF-IDF: It considers that if "more specific words" in a given sentence, then the sentence is relatively more important. Comparison between the term frequency (tf) in a document (in this case each sentence is treated as a document) and the document frequency (df).

$$\frac{TF}{IDF(w)} = DN\left(\frac{\log(1+tf)}{\log(df)}\right)$$

where DN is number of documents.

Example: Consider a document containing 100 words wherein the word 'cat' appears 3 times. Then, Tf(cat)=(3 / 100) = 0.03. Now, assume we have 10 million documents and the word 'cat' appears in one thousands of these. Then, Idf(cat)= log(10,000,000 / 1,000) = 4. Thus, the Tf-idf = 0.03 * 4 = 0.12.

iii. Upper Case: This method assigns higher scores to words that contain one or more upper case letters. It can be a proper name, initials, highlighted words, among others. The score is calculated as:

$$CPTW(j) = \frac{NCW(j)}{NTW(j)}$$

where: CPTW = Ratio of total first letter capital words present in the sentence to the total number of words present in the sentence NCW = Number of first letter capital words NTW = Total number of words present in sentence.

Uppercase value feature:

$$UCf = \frac{CPTW(j)}{MAX(CPTW(j))}$$

Proper Noun: Usually the sentences that contain a higher number of proper nouns are more important; thus, they are likely to be included in the document summary. This is a specialization of the Upper-case method.

Word co-occurrence: Word co-occurrence measures the chance of two terms from a text appear alongside each other in a certain order. One way to implement this measure is using n-gram, which is a contiguous sequence of n items from a given sequence of text or speech. In short, it gives higher scores to sentences that co-occurrence words appear more often.

Lexical similarity: It is based on the assumption that important sentences are identified by strong chains. In other words, it relates sentences that employ words with the same meaning (synonyms) or other semantic relation.

B. Sentence Scoring

This approach analyzes features of the sentences like Cue-phrases (domain specific phrases), inclusion of numerical data, length of sentences, position, centrality, resemblance with title, etc.

i. Cue Phrases: Analysing the presence of cue words in sentences They include the sentences started by "in summary", "in conclusion","our investigation", "the paper describes" and emphasizes such as "the best", "the most important", "according to the study", "significantly", "important", "in particular", "hardly", "impossible" as well as domain-specific bonus phrases terms can be good indicators of significant content of a text document . A higher score is assigned to sentences that contain cue words/phrases, using the formula:

$$CP = \frac{CPS}{CPD}$$

where, CP = Cue-phrase score,
CPS = Number of cue-phrases in the sentence,
CPD = Total number of cue-phrases in the document in equation.

ii. Sentence inclusion of numerical data: Usually the sentence that contains numerical data is an important one and it is very likely to be included in the document summary, according to references. This kind of sentence usually refers to some important information such as date of event, money transaction, damage percentage, etc.

iii. Sentence Length: This feature is employed to penalize sentences that are too short or too long, these sentences are not considered as an optimal selection. The method uses length as number of words in sentence. In addition, sentences can be penalized that are shorter than certain length. The first case could be calculated as follows:

$$Score = Length(s) * AverageSentenceLength$$

iv. Sentence position: There are many approaches that use the sentence position as a score criterion.

v. Sentence centrality: Sentence centrality is the vocabulary overlap between a sentence and other sentences in the document. Centrality could be calculated as follows:

$$Score = \frac{Ks \cap KOs}{Ks \cup KOs}$$

where, Ks = Keywords in s KOs = Keywords in other sentences

vi. Sentence resemblance to the title: Sentence resemblance to the title is the vocabulary overlap between this sentence and the document . In this case, sentences similar to the title and sentences that include the words in the title are considered important. way to calculate this score is:

$$Score = \frac{Ntw}{T}$$

where,

Ntw = Number of title words in sentence

T = Number of words in the title.

C. Graph Scoring

In this method, score is calculated on the basis of relationship found among the sentences. It includes TextRank model, Bushy path, Aggregate similarity algorithms, etc. When a sentence concerns to another it forms a link with an allied weight between them. The weights are used to obtain the score of sentences. Bushy path of a node: In this method we find number of links connecting it to other nodes.

Aggregate Similarity: This method uses same approach as bushy path of a node but it sums the weights (similarities) on the links. TextRank Algorithm: TextRank is an algorithm based upon PageRank for text summarization. In TextRank, the vertices of the graph are sentences, and the edge weights between sentences denotes the similarity between sentences. The TextRank Algorithm is described below:

Algorithm:

1) Obtain the text units that best define the job at hand, and add those units as vertices in the graph.

2) Find relations that connect such text units to draw edges as connectors between those vertices.

3) Iterate the graph-based ranking algorithm until all possible connections are achieved.

4) Classify vertices based on their final obtained score. Utilize the values attached to each vertex for ranking/selection purpose.

5)Consider, $G = (V, E)$ be a digraph with the set of vertices $V$and set of edges E. Where, $In(V_i)$ the set of vertices V_i that point to it (predecessors), and $Out(V_j)$set of vertices V_j that points to (successors). Then, score can be calculated as:

$$S(V_i) = (1 - d) + d * \sum_{j \in In(V_i)} \frac{S(V_j)}{Out(V_j)}$$

and d is a damping factor that can be set between 0 and 1, which has the role of combining into the model the probability of moving from a given vertex to another arbitrary vertex in the graph. The factor d is traditionally set to 0.85 [17]

We can develop an extractive Marathi summarizer using above feature extraction and sentence scoring techniques. Among which Graph based model TextRank can be used due to its linguistics independence and efficiency.

VI. EVALUATION MEASURES

Evaluation can be done Qualitatively or Quantitatively based on the methods used. Qualitatively, the objective is to find a summary grammatically and semantically correct, that is relevant, and that the user can approve (as opposed to disapprove) and/or give a score which constitutes an accepted summary (as opposed to a rejected summary proposal). Quantitatively, the most used way to evaluate the factualness of text summaries is to compare them with human-made summaries. These summaries will be compared with the output of our program using the ROUGE metric.

ROUGE i.e. Recall-Oriented Understudy for Gisting Evaluation, is a set of metrics and a software package used for evaluating automatic summarization and machine translation software in natural language processing. The metrics compare an automatically produced summary against a reference or a set of references (human-produced) summary. In our work we have compared the system generated summaries with two sets of human generated summaries. Based on the n-gram overlap between candidate and reference summaries ROUGE calculates the scores of a candidate summary [18].

Based on n-grams used in the evaluation ROUGE-N consists of different metrics, such as ROUGE-1, ROUGE-2, ROUGE-3 and so on. N varies between 1-4. The recall (R) measure is calculated by the proportion of n-grams from reference summaries occurring in a candidate summary, the precision (P) is calculated by the proportion of n-grams from a candidate summary occurring in reference summaries and F-score can be then calculated by combining recall and precision into one metric.

$$Recall = \frac{|G_{ref} \cap G_{can}|}{|G_{ref}|}$$

$$Precision = \frac{|G_{ref} \cap G_{can}|}{|G_{can}|}$$

ROUGE-N scores are computed as follows:

$$ROUGE_{F-score} = \frac{2 * ROUGE_{recall} * ROUGE_{precision}}{ROUGE_{recall} + ROUGE_{precision}}$$

where, G_{ref} refers to the grams of reference summary and G_{can} refers to candidate summary.

Our work for constructing one such Marathi extractive text summarizer combining positional and similarity approach with TextRank gives good results. The generated f-scores are shown in table below:

Table 1. F-Scores

Technique	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4
TextRank+ Positional	0.8004	0.7909	0.785	0.7799
TextRank+ Similarity	0.8684	0.8602	0.8521	0.8447

Figure 2 shows plotting of F-scores with ROUGE-N results for N=1-4.

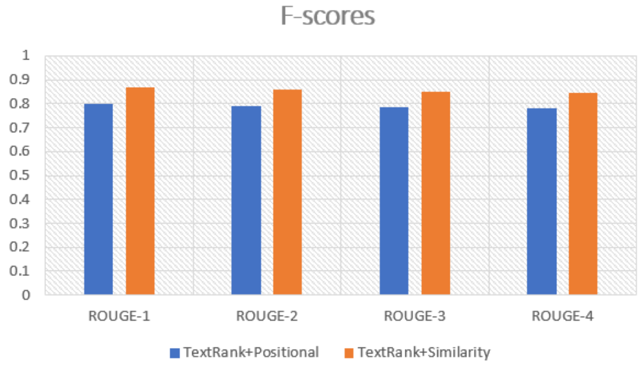

Figure 2. ROUGE Results

VII. CONCLUSION

With the recent increase in the amount of content available online, fast and effective automatic summarization has become more important. The most important steps in this system approach are feature extraction, scoring and graph generation. This system can be used in various fields like education, in search engines to improve their performances, for Marathi news clustering, Question generation purpose and many other application oriented areas, etc.

VIII. REFERENCES

[1]. Virat V. Giri, Dr.M.M. Math and Dr. U. P. Kulkarni, A Survey of Automatic Text Summarization System for Different Regional Languages in India Bonfring International Journal of Software Engineering and Soft Computing, Vol. 6, Special Issue, October 2016

[2]. Sheetal Shimpikar and Sharvari Govilkar, A Survey of Text Summarization Techniques for Different Regional Languages in India, International Journal of Computer Applications, Vol. 165, No. 11, May 2017

[3]. Sunitha C, Dr. A Jaya and Amal Ganesh, A Survey of Abstractive Summarization Techniques in Indian Languages, 2016

[4]. Hamzah Noori Fejer and Nazlia Omar, Automatic Multi-Document Arabic Text Summarization Using Clustering and Keyphrase Extraction ICIMU IEEE 2014 International Conference,978-1-4799-5423-0.

[5]. Deepali K. Gaikwad, Deepali Sawane and C. Namrata Mahender, Rule Based Question Generation for Marathi Text Summarization using Rule Based StemmerIOSR Journal of Computer Engineering (IOSR-JCE), eISSN: 2278-0661. 2015 6Mudassar Majgaonkar and Tanveer Siddiqui, Discovering suffixes: A case study for Marathi Language (IJCSE) International Journal on Computer Science and Engineering Vol. 02, No. 08, 2010, 2716-2720

[6]. Aishwarya Sahani, Kaustubh Sarang, Sushmita Umredkar, and Mihir Patil, Automatic Text Categorization of Marathi Language Documents (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 7 (5) , 2016, 2297-2301

[7]. Ms. Jayshri Arjun Patil, Ms. Poonam Bhagwandas Godhwani, Review of Name Entity Recognition in Marathi Language IJSART - Volume 2 Issue 6 , June 2016

[8]. A report on Text Summarization for Compressed Inverted Indexes and snippets by Mahesh Dangale CS 297 Report July 2013.

[9]. Anjali R. Deshpande, Lobo L. M. R. J., Text Summarization using Clustering Technique International Journal of Engineering Trends and Technology (IJETT) - Volume4 Issue8-August 2013

[10]. Manpreet kaur, Usvir Kaur, Comparison Between K-Mean and Hierarchical Algorithm Using Query Redirection International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 7, July 2013

[11]. Feifan Liu, Yang Liu, Exploring Correlation between ROUGE and Human Evaluation on Meeting Summaries IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING

[12]. R. Vijaya Lakshmi, Dr. S. Britto Ramesh Kumar, Literature Review: Stemming Algorithms for Indian and Non-Indian Languages International Journal of Advanced Research in Computer Science & Technology IJARCST Volume 2, Issue 3, July-Sept 2014

[13]. Pallavi Bagul, Archana Mishra, Prachi Mahajan, Medinee Kulkarni, Gauri Dhopavkar, Rule based POS tagger for Marathi TextInternational Journal of Computer Science and IT technologies, Volume 5, 2014

[14]. Pooja Pandey, Dhiraj Ahim, Sharvari Govilkar, Rule based Stemmer using Marathi WordNet for Marathi Language International Journal of Advanced Research in Computer and Communication Engineering, Volume 5, Issue 10, 2016

[15]. Rafael Ferreira, Luciano de Souza Cabral, Assessing sentence scoring techniques for extractive text summarizationExpert Systems with Applications, Elsevier 2013

[16]. TextRank: Bringing order into texts, Mihalcea, Rada., and Tarau, Paul. (2004), In Conference on empirical methods in natural language processing, Barcelona, Spain.

[17]. Rouge: A package for automatic evaluation of summaries, Lin, C. Y. In Text summarization branches out, Proceedings of the ACL-04 workshop (Vol. 8).

[18]. "Variations of the Similarity Function of TextRank for Automated Summarization", Federico Barrios, Federico Lopez, Luis Argerich, Rosita Wachenchauzer, arXiv, 2017.