Graph Based Technique for Hindi Text Summarization

Conference Paper · January 2015

DOI: 10.1007/978-81-322-2250-7_29

3 authors, including:

Vimal Kumar K
Jaypee Institute of Informati…
4 PUBLICATIONS **13** CITATIONS
SEE PROFILE

Divakar Yadav
Jaypee Institute of Informati…
48 PUBLICATIONS **85** CITATIONS
SEE PROFILE

Graph Based Technique for Hindi Text Summarization

K. Vimal Kumar, Divakar Yadav and Arun Sharma

Abstract Automatic Summarization is the process of generating or extracting the important sentences from the given input document. Since there are many such systems for English language so this proposed system is mainly focused on the Hindi language. The basic idea of this summarization system is to identify the important sentences and also to extract them based on its relevance with other sentences. In case of summarization the sentences in the summarized document should be meaningful and relevant to each other, which are achieved using sentential semantic analysis. For finding the relation between each sentence and also to analyze for the importance, the Graph based approach is found to be more appropriate. Based on the frequency of words occurrence in the input document, the sentences are ranked and the ranks are used to identify the important sentences in the document. The relevance between each sentence in the document with other sentences is found using semantic similarity. There may be same information conveyed by two different sentences whose semantic similarity score is very high. Such kind of sentences has to be kept only once in the output. For which an analysis has been performed over various semantically similar sentences. Finally, the identified relevant sentences are merged using the rank and the semantic analysis of the sentences. These identified sentences are rearranged to provide a proper meaningful summarized text to avoid textual continuity in the output text. The system is found to perform well in terms of precision, recall and F-measure with various input documents.

Keywords Extractive approach · Graph approach · Hindi language · Summarization · Semantic analysis · Latent semantic analysis

K.V. Kumar (✉) · D. Yadav
Jaypee Institute of Information Technology, Noida, India
e-mail: vimalkumar.k@gmail.com

D. Yadav
e-mail: divakar.yadav@jiit.ac.in

A. Sharma
Gautam Buddha University, Greater Noida, India
e-mail: arun08sharma@gmail.com

J.K. Mandal et al. (eds.), *Information Systems Design and Intelligent Applications*,
Advances in Intelligent Systems and Computing 339,
DOI 10.1007/978-81-322-2250-7_29

1 Introduction

Automatic text summarization system produces a concise about the input document by retaining the most important and relevant text in the output text. Since the information overload on the web has increased the problem of decision making, the need of such a summarization system has increased. These type of summarization systems helps in making decisions as it provides the compressed form of the complete document. There are various other applications which use summarization system and one such application is on search engines to provide abstract information about various links that are searched by the end user. The accuracy of the decision making and search engines depends more on the accuracy of the summarization system being used in it as this is the back bone for those kinds of systems. The summarization system can be designed using two approaches namely Extractive and Abstractive approach. Extractive approach basically extracts out the important and relevant words or phrases or sentences from the input document. Whereas the abstractive generates the output text based on its semantic understanding of the input text. The automatic text summarization system assists in decision making, question answering system and so on. For these kinds of system, there is need for one or more input document to improve the efficiency. There comes the need for multi document summarization system. In case of multi document summarization system, the same extractive and abstractive approach can be used. But, before using these approaches, the system has to identify the relevant documents from the set of input documents. The identification of relevant documents is totally application specific. Considering the descriptive question answering system which requires summarization of the answer based on the input question and these kinds of system also requires a set of documents on different domain so that answers can be retrieved irrespective of the domain. The documents that matches this input question has to be identified and the entire identified documents will be subjected to any of the extractive or abstractive approach to generate the required output text.

This proposed system is designed in such a manner that the summarized text provides the important and relevant information provided in the input text so that there is no loss of information during compression. The graph theoretic approach used in this research makes use of extractive approach to identify important sentences. This extractive approach makes use of normalized TF-IDF to rank the sentences. A TF-IDF matrix has been created between words that are not stop words. To eliminate the stop words, a preprocessing stage is included which makes use of stop words list. After elimination of stop words, the main words are converted to vectors based on the frequency of occurrence of words in each and every sentence of the input document. The TF-IDF values of these words are normalized in the sentential level. Based on the normalized values, the sentences are ranked according to their importance. Now the system has the important sentences but may not be meaningful as there won't be related sentences in the output. So the relevance of extracted important sentences has to be found for which the semantic analysis is carried out between these sentences and edges are created if there is

semantic relevance between the sentences. The semantic analysis gives the similarity between sentences based on its meaning which can provide the information about the relevant text that are required for the summarization system. The semantic analysis used in this system is based on the widely used Latent Semantic Analysis (LSA). These important and relevant sentences have to be put in the output but may not be on the same order of occurrence in the input file. There is need for rearranging the sentences based on their order of occurrence in the input text.

The rest of paper has been organized as: Sect. 2 is about the various relevant works that has been made by various researchers in this particular area. The Sect. 3 describes about the proposed graph based approach in detail with the mathematical explanation. Section 4 describes about the analysis carried over on this system with various input text and also deals about the efficiency of this system. The Sect. 5 deals with the drawbacks of this system and how it can be eradicated in the future.

2 Related Work

In recent years, there is vast growth in the number of research works on the automatic text summarization and are focused mainly on the extractive approach. There are systems which make use of hand crafted rules made through various templates that are used to identify the important sentences. This method is confined to the application or on the input document under consideration. For example, the sentences which occur in the abstract and conclusion part of any research document are more important compared to other sections. In the question answering systems, the sentence which matches with the question are more important sentences which has to be retained in the output extracted text. These kinds of rule based system can be improved by unsupervised or supervised algorithms [1–3].

The semantic analysis of sentences can be used to extract out the sentences. For which various algorithms are available such as Latent Semantic Analysis (LSA), Point Mutual Information (PMI), Lesk algorithm. Pal et al. has proposed the Wordnet based method to identify the semantics behind various input text by making use of Lesk algorithm. Based on the semantic analysis, the important sentences are identified and extracted [4].

Devasena et al. has mentioned about a rule based approach for text categorization and summarization. The text categorization is performed using set of preclassified examples. Once the sentences are classified, the sentences are included in the summarized document if the set of rules are matched. This system has provided good results but may not work in case of sentences not mentioned in the example text and also is limited to the set of rules included in the system [5].

A star map is used in summarization system developed by Kalaiselvan and Kathiravan [6], which constructs a star map between identified important and meaningful sentence based on the linguistic and statistical features of sentences. This system is found to have various applications such as duplicate elimination, exam paper evaluator, lesson planning, Identify Shingling.

Personalized summarization system has improved the conventional summarization system by taking into account the readers preferences. The main source of identifying the reader's preference is through annotation of the documents. Moro and Bielikov' [7] has mentioned that the system performs well by considering the domain in which the text is being used based on the reader's choice. The personalized system may not perform well for all domains as there are words having different interpretation in different domains. For example, the word TABLET has different senses in the electronics and medical domain.

Another approach is to use Universal Networking Language based approach [8]. It is used basically for a language independent application system. A multi lingual summary can be generated with much more ease by using a interlingua document representation language called, "Universal Networking Language" (UNL). This approach can be applied over other languages as well.

3 Proposed Method—A Graph Based Approach

Graph based approach is basically designed to provide the summarized text by identifying important, relevant and informative text from the input text for the compression ratio selected by the end user. This approach is broadly divided into three phase—Sentence ranking, Sentential semantic analysis and Sentence extraction (as shown in Fig. 1).

In brief, the sentence ranking algorithm is applied on the input sentences to identify which are important sentences and the relation between each sentence is calculated using the semantic analysis. The following are the description in detail about these phases of the graph based approach,

3.1 Sentence Ranking

In this phase, the system identifies the important sentences based on the sentence ranking method. Sentence ranking is based on the frequency of occurrence of various words mentioned in the text. The TF-IDF method is identified to be well suited to achieve this task. Usually, TF-IDF is applied over a set of document, but since this system is designed over a single input document, the normalized term frequency is considered for sentence ranking. The normalized term frequency is applied over the words mentioned in the texts. So before applying this ranking method, the tokenization has to be made, both at sentence level and word level. With the help of this tokenization process, the input sentence is tokenized to identify each word present in input text [9, 10]. In case if all the input tokenized words are considered for ranking, the rank of sentences will be very biased based on the number of occurrence of various stop words such as Hindi conjunctions. For example, the input sentences may have frequently occurring words such as का, की, etc. and these words may bias the

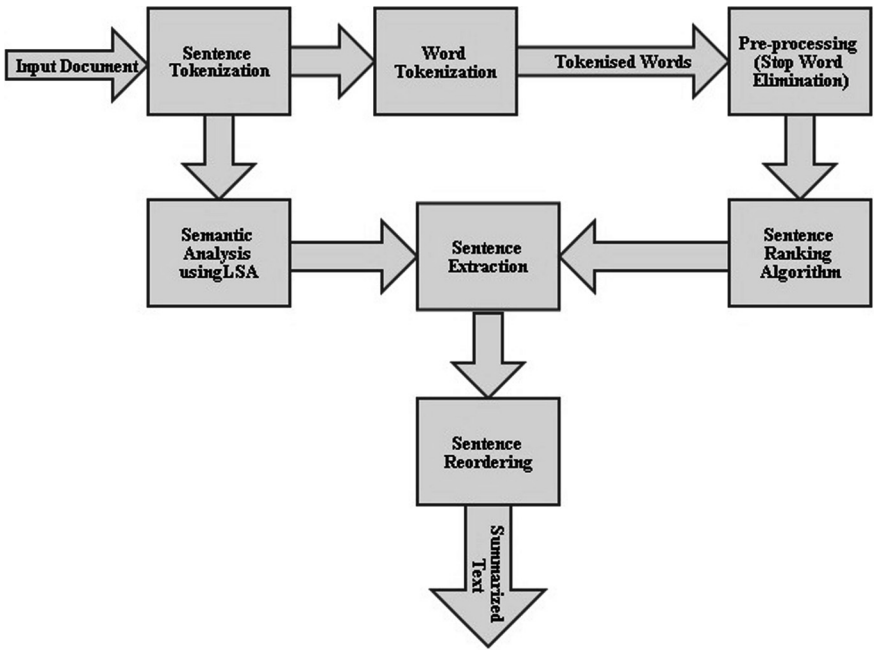

Fig. 1 System architecture

ranking strategy. To avoid such kind of biasing, a preprocessing of the input sentence to eliminate the stop words has to be carried out. A list of Hindi stop words has been made for this preprocessing stage. The term frequency of these preprocessed words is stored in a matrix and the normalized TF is applied over each term mentioned on each sentence to rank the sentences. The normalized term frequency is defined as follows,

$$TF_{norm} = \alpha + (1 - \alpha) * \frac{tf}{tf_{\max}} \tag{1}$$

where, α ranges from 0 to 1 and ideally its set as 0.4.

Based on the sentential value of normalized term frequency, the sentences are ranked and are identified as important sentences in the input text. The system needs to detect the relevant sentences among these important sentences which are identified in the semantic analysis phase.

3.2 Sentential Semantic Analysis

The latent semantic analysis (LSA) is used to perform semantic analysis for which the GENSIM tool has been made use of. During sentential semantic analysis the

system should have a grammatically valid sentence so; the system considers the input text as it is by neglecting the preprocessing stage which was used in the previous phase. Then semantic analysis is carried over between every other sentence mentioned in the input document. LSA is basically defined as an application of singular value decomposition over a vector. Initially, each sentence is converted to its corresponding term frequency vectors and these vectors are decomposed into three different matrices—left singular vectors (L), right singular vectors (R) and singular diagonal vector (S) (as shown in Eq. 2),

$$\begin{bmatrix} f_{t1}^{1} & f_{t2}^{1} & \cdots\cdots & f_{tn}^{1} \\ f_{t1}^{2} & f_{t2}^{2} & \cdots\cdots & f_{tn}^{2} \\ \cdots & \cdots & \cdots & \cdots \\ f_{t1}^{m} & f_{t2}^{m} & \cdots\cdots & f_{tn}^{m} \end{bmatrix} = L * R * S \tag{2}$$

where f_{tn}^{m}—indicates the frequency of nth-term in mth-sentence.

The term frequency of words is arranged in the form of a matrix with column as different terms (words) used in the text and row of the matrix as the sentence identification numbers. After the decomposition of this matrix, the product of the right singular matrix and singular diagonal matrix will give the frequency of semantically similar words in different sentences and the similarity between two sentences is found using cosine similarity over the product of right singular vector and singular diagonal vector of the sentence considered. The cosine similarity is calculated using the below mentioned in the equation,

$$\cos\theta = \frac{\sum_{i=1}^{n} x_i * y_i}{\left(\sqrt{\sum_{i=1}^{n} x_i^2} * \sqrt{\sum_{i=1}^{n} y_i^2}\right)} \tag{3}$$

If any of the two sentences is found to be semantically similar, an edge is created between those two sentence nodes. At this phase, the system has identified the relevant sentences and in the previous phase the important sentences are identified, now the system has to extract out the sentences based on the compression ratio mentioned by the end user which is performed in the sentence extraction phase.

3.3 Sentence Extraction

The identified sentences are extracted based on the rank and its relevance with other sentences [11, 12]. Visualizing the sentences as nodes in a graph and connecting those nodes which are containing semantically similar sentences. The node which has highest rank is selected and all its relevant sentences are also extracted to the output set of sentences except the sentences which matches above 90 %. Based on

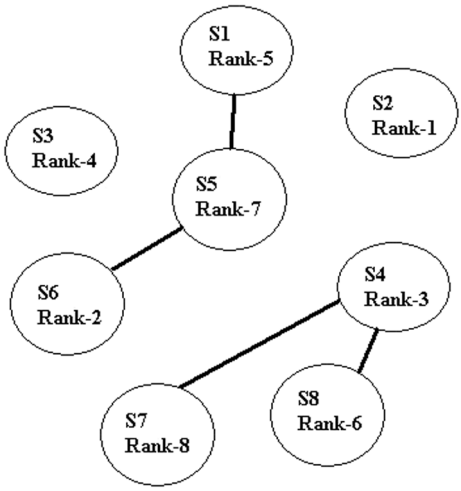

Fig. 2 Graph representation of sentences

analysis, the sentences whose semantic similarity is more than 90 % are found to convey the same information. To avoid redundant information in the output text, these sentences are not duplicated. In a decreasing order of rank, each of the sentences are considered and extracted to the output text along with its set of semantically related sentences by keeping an eye on the compression ratio.

Considering eight set of sentences, the graph representation according to the analysis performed in previous phase will be as shown in Fig. 2. In Fig. 2, S1 represents sentence-1 and an edge between two nodes denotes that there is semantic relationship between them. Applying the sentence extraction phase over these sentences shown in Fig. 2 will extract the sentence S2 (having rank-1) and there is no semantically similar sentence which is indicated by no connecting edge from this particular node. It'll extract the next highest ranked sentence, i.e., S6. Now, the sentence S6 is connected to S5, it extracts S5 as well. This extraction process is continued until the desired compression is reached. In case a node such as S4 which has two nodes connected with it, then the one which has highest rank among them is considered.

These extracted sentences do not make any proper sense for the final output. So, there is a need for sentence reordering in which the sentences are rearranged to the order of occurrence in the input document. Thus the summarized output will have proper sense without losing the relevant and important information conveyed in the input document.

4 Results and Discussion

This system has been tested and analyzed on various documents selected from different domains at different compression ratio. The analysis (mentioned in Table 1) has shown that the system's recall measure is found to decrease with

Table 1 Precision, recall and F-measure comparison

Compression ratio (%)	Precision (average) (%)	Recall (average) (%)	F-measure (average) (%)
40	76.79	80.53	80
60	79.42	68.78	70
80	82.89	45.27	60

increase in compression ratio. But, the precision of the system is found to improve with increase in compression ratio. On an average the system performs well for the 60 % compression ratio. The systems performance also depends on the input document chosen.

Table 2 and Fig. 3 shows the precision and recall of the system for 60 % compression ratio over five different documents. This analysis shows that the system outperforms in case of 60 % compression ratio and gives an average F-measure as 70 %. The system is also found to perform well with 80 % compression ratio but 60 % compression ratio is considered as an average performance. The degrade in performance at 80 % compression ratio is found to be because of restriction to have less number of output text to convey the same amount of information given in original input text.

Table 2 Comparison for various documents

Document ID	Precision (%)	Recall (%)	F-measure (%)
1	78.26	56.52	70
2	83.34	77.78	80
3	77.78	66.67	70
4	84.61	76.92	80
5	77.78	77.78	80

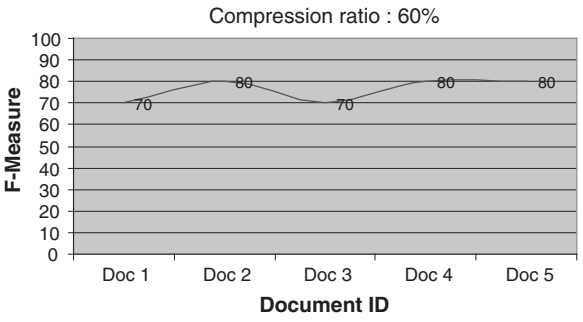

Fig. 3 F-measure comparison of various documents

5 Conclusion and Future Work

On average the system's precision, recall, F-measure are calculated as 79, 69 and 70 % respectively. The system gives higher recall which indicates that the system conveys 69 % of the required relevant information from the input text. Also the improvement in precision indicates that the text retrieved by system matches 79 % of with the idle summaries for those documents. So to conclude, this summarization system is found to give improved results in terms of improved precision, recall and F-measure.

The downfall in performance at higher compression ratio in this system can be improved by merging the similar sentences to convey the information on different sentences in a single merged sentence. By including a module to merge those sentences that are found to be more similar or neglecting those sentence those doesn't convey much information can improve this system a lot. Also, this system can be improvised further to increase the precision and recall by applying an abstractive approach over the extractive one. By applying abstractive approach to identify the important sentence and using the semantic analysis to identify the linkages can improve the system. This system can also be used for multi document summarization and can be extended to any of the applications that make use of summarization system.

References

1. Juneja, V., Germesin, S., Kleinbauer, T.: A learning-based sampling approach to extractive summarization. In: Proceedings of the NAACL HLT 2010 Student Research Workshop, pp. 34–39 (2010)
2. Gupta, V., Lehal, G.S.: A survey of text summarization extractive techniques. J. Emerg. Technol. Web Intell. **2**(3), 258–268 (2010)
3. Gupta, V., Lehal, G.S.: Features selection and weight learning for punjabi text summarization. Int. J. Eng. Trends Technol. **2**(2) (2011)
4. Pal, A.R., Saha, D.: An approach to automatic text summarization using WordNet. In: Advance Computing Conference (IACC), 2014 IEEE International, pp. 1169, 1173 (2014). doi:10.1109/IAdCC.2014.6779492
5. Devasena, C.L., Hemalatha, M.: Automatic text categorization and summarization using rule reduction. In: Advances in Engineering, Science and Management (ICAESM), 2012 International Conference on, pp. 594, 598, 30–31 Mar 2012
6. Kalaiselvan, M., Kathiravan, A.V.: A pioneering tool for text summarization using star map. In: Pattern Recognition, Informatics and Mobile Engineering (PRIME), 2013 International Conference on, pp. 277, 281, 21–22 Feb 2013
7. Moro, R., Bielikov', M.: Personalized text summarization based on important terms identification. In: Database and Expert Systems Applications (DEXA), 2012 23rd International Workshop on, pp. 131, 135 (2012) doi:10.1109/DEXA.2012.47
8. Mangairkarasi, S., Gunasundari, S.: Article: semantic based text summarization using universal networking language. Int. J. Appl. Inf. Syst. **3**(8), 18–23 (2012) (Published by Foundation of Computer Science, New York, USA)
9. Porter, M.F.: An algorithm for suffix stripping. Program **14**(3) (1980)

10. Ramanathan, A., Rao, D.D.: A lightweight stemmer for Hindi. In: Proceedings of EACL (2003)
11. Alguliev, R.M., Aliguliyev, R.M.: Effective summarization method of text documents. In: Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence (WI'05), pp. 1–8 (2005)
12. Mihalcea, R., Tarau, P.: An algorithm for language independent single and multiple document summarization. In: Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP), Korea (2005)