International Journal for Research in Engineering Application & Management (IJREAM)
ISSN : 2454-9150 Vol-04, Issue-10, Jan 2019

Automatic Summarization of Hindi Text Documents Using Supervised Learning Method

[1]Dipali Telavane, [2]Apurva Khude, [3]Kartik Lakade, [4]Mohini Chaudhari
[1,2,3]Research Scholar, [4]Assistant Professor, Vidyalankar Institute of Technology, Mumbai, India.
[1]dipalitelavane@gmail.com, [2]apurvakhude@gmail.com, [3]kartiklakade@gmail.com,
[4]mohinichaudhari89@gmail.com

***Abstract*: The availability of information today accessible in digital form has accelerated. Retrieving useful document from such large pool of information gets difficult. So, to summarize these text documents is very crucial. Text summarization is a process of minimizing the original source document to get essential information of that document. It eliminates the redundant, less important content and provides you with the vital information in a shorter version usually half a length of the original text. Creating a manual summary is a very time-consuming task. Automatic summarization helps in getting the gist of information present in a particular document in a very short period. In the comparison of all Indian regional languages, there is very less amount of work done for summarization of Hindi documents. This paper presents an effective way to summarize Hindi text documents using a supervised learning method based on extractive summarization approach. It focuses on summarizing single Hindi text document at a time based on natural language processing (NLP). Multiple features are extracted from every sentence in a document. Sentences are scored based on the features extracted. Depending on the scores of sentences, sentences with higher score are selected for summary.**

Keywords*: *Extractive, Feature extraction, Hindi Text Summarization, Naïve-Bayes classifier

I. INTRODUCTION

Internet exchanges a huge amount of data. In this new era, the Internet is being proliferated. So the problem of information overload has increased. Rather than reading the entire document which includes many examples, comparisons, supporting details, etc. for the readers, it is always convenient to read point to point specific gist of the document. Automatic text summarization is actually meant for this. This gives the reader a non-redundant presentation of the facts found in filtered details of the source text. Automatic text summarization is a process that extracts the most essential part of the original source document. It eliminates unnecessary, less important content and provides the essential information in a small version normally half a length of the original text. Summarization can be divided into two text categories: Extractive Summarization vs. Abstractive Summarization [7].

Extractive Summarization: An extractive summarization is a process which selects few sentences from all the sentences in a document to be included in the summary. In this process, every sentence is ranked using different features of that sentence. On the basis of ranks, sentences to be added in the summary are decided. In extractive summarization, features of sentences are extracted using linguistic and statistical analysis.

Abstractive Summarization: Abstractive summarization is a process which understands the original text in short and rephrases it in the fewer words. In abstractive summarization, semantic analysis of the document is done. In process, after resolving each of the sentences, sentences may be represented in the different formats by reformulating the original one. Generally, the abstractive summarization process is more complex as it involves the understanding of the document and recreating summary using new notions and terms. But this method of summarization is more efficient when compared with manmade summaries.

Majority of the research work done so far has been more emphasis on widely used English and other European languages. Due to the low volume of information available in the non-English language, the Indian Languages have been explored little less [9]. However, the scenario is changing now and information in large quantities is available in different languages. The need for text summarization methods that handle Indian languages has seen growth. Hindi is written using Devanagari script and it has its own alphabet set. A Hindi root word consists of many morphological variants that are associated with

 DOI : 10.18231/2454-9150.2018.1408

inflection, making it difficult to extract a feature from Hindi texts.

In our study, we propose a system for automatic extractive summarization techniques for the creation of a summary of Hindi text documents. The system would currently produce a summary for single text documents at a time which are in Hindi. Hindi text summarization has various applications in those systems where text analysis and knowledge representation are required. This system is based on extractive summarization approach. It attempts to identify the set of key sentences that are most important to understand a given document in Hindi. Accuracy of the system majorly depends on how correctly significant sentences are selected from the document. Whenever a document is given as input, pre-processing is done on it to represent a document in a controlled manner required for the further purpose. Now features related with every sentence are calculated and scores are given to sentences. Most ranked sentences are selected for the final summary. For scoring the sentences, supervised learning methods such as Naïve-Bayes are used. Naïve-Bayes classifies sentences as to be selected in summary or not. Sentences selected by using Naïve-Bayes are further used for generation of a summary.

This paper is structured into 5 sections. Section 1 presents the introduction; section 2 describes Literature Survey of text classification. Section 3 describes the proposed system of extractive text summarization for Hindi documents. Section 4 presents the methodology and finally, the conclusion is included in section 5.

II. LITERATURE SURVEY

Here the relevant literature survey that uses various techniques for text summarization of Indian language documents is presented. Most of the researchers focused on supervised machine learning techniques for summarization of Indian regional language documents. These techniques provide better results in the form of accuracy and time efficiency.

Chetana Thaokar and Latesh Malik [1] presented an idea to summarize Hindi text using sentence extraction method. Hindi WordNet was used in the system to tag POS of words for checking Subject-Object-Verb (SOV) of the sentence. Later they used the genetic algorithm to optimize the process of summary generation and to minimize redundancy. At the end sentence ranking was used to rank the sentences according to the importance of them in their summary.

Dawinder Kaur, Rajbhupinder Kaur [2] developed a rule-based approach to generate the summary from a text document written in the Hindi language. For that, they have taken Hindi text paragraph and first perform pre-processing on it. In the next step, they assign the weights o the sentences according to the features present in sentences. Later they eliminate sentences having a lesser weight than that of minimum weight. In the end, they combine the other lines as a paragraph. They have tested on 60 documents from a different domain.

Vishal Gupta and Gurpreet Singh Lehal [3] have designed an Automatic Punjabi Text Extractive Summarization system. This system consists of two phases 1) Pre Processing 2) Processing. Pre-processing is done to distinctly identify sentences and words and remove Punjabi stop words elimination. The processing phase calculates sentence features. Sentences are scored based on features. Depending on the score, not important sentences are eliminated from the input text. It is described that the author tested the proposed system over fifty Punjabi news documents and fifty Punjabi stories.

Madhi A. Ali, A. Al-Dahoud and Bilal H.Hawashin [4] proposed an intelligent system for text summarization of English documents. They used the sentence ranking method to score each sentence. Depending on the score, the sentences to be selected in the final summary are decided. They used three supervised learning methods, mainly SVM, Naïve-Bayes, and decision tree classifier.

Bijal Dalwadi, Nikita Patel and Sanket Suthar [5] reviewed different text Summarization techniques available for Indian languages. They describe the process of extracting the important information and reducing the size of the original document and producing a summary by retaining important information on the original document.

Vipul Dalal and Dr. Latesh Malik [6] have created an automatic abstractive text summarization system for Hindi documents. They first performed pre-processing on a document to clean it. Later they calculate sentence features. They used the approach of bio-inspired computing. To create a summary semantic graph and particle swarm optimization algorithm was used.

Saiyed Saziyabegum and Priti S. Sajja [7] created a literature review on different approaches available for extractive summarization of any language document. They suggest various methods to do the extractive summarization of single documents. They also suggest methods to do the extractive summarization of Multilanguage document or summarization of multiple documents at once.

III. PROPOSED SYSTEM

The proposed system is automatic summarization of Hindi text documents using supervised learning methods. The input to the system is Hindi text documents and result i.e. output is an extractive summary of Hindi documents. Extractive summary is obtained by identifying the

 DOI : 10.18231/2454-9150.2018.1408

important sentences from the input text. The importance of sentences is determined based on different features of sentences which are extracted by feature extraction method [9]. The proposed method uses Naïve-Bayes to decide either the sentence to be involved in summary or not.

1. Read the input document
2. Pre-processing of document
   - 2.1. Segmentation
   - 2.2. Tokenization
   - 2.3. Stop word Removal
   - 2.4. Stemming
3. Feature Extraction
   Extract the following features from each sentence.
   - 3.1. Sentence Paragraph Position (f1)
   - 3.2. Sentence Overall Position (f2)
   - 3.3. Numerical Data in Sentence (f3)
   - 3.4. Presence of Inverted Commas (f4)
   - 3.5. Sentence Length (f5)
   - 3.6. Keywords in Sentence (f6)
   - 3.7. Title similarity (f7)
   - 3.8. Term Frequency-Inverse Sentence Frequency (f8)
   - 3.9. Sentence to sentence similarity (f9)
4. Sentence ranking

Apply Naïve-Bayes model to rank the sentences from 4 to 0 where sentence with ranking 4 is considered as the most important sentence and sentence with ranking 0 is considered as the least important sentence.

5. Generate summary

Extractive type of summary of given input document is generated and given as output to the user.

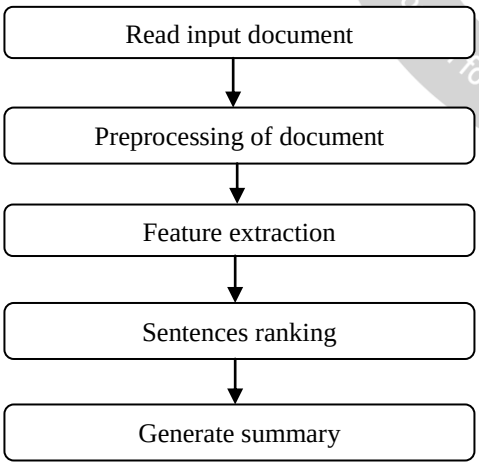

Fig. 1. Block diagram of automatic extractive summarization system

IV. METHODOLOGY

The system takes input as Hindi text documents. First the pre-processing is done on the documents. Pre-processing step includes input validation, tokenization, stop word removal and stemming. Then the features are extracted from pre-processed tokens. On the basis of features, summary of document is created using supervised machine learning methods.

Following the main steps are involved in extracting Hindi documents:

1. Preprocessing step

In this step, a Hindi document is taken as an input. The document is processed to obtain it in a controlled format and remove redundancy and improve the efficiency of summarization as well.

1.1. Segmentation –

The input document proceeds by checking input Hindi text into the proper format. Character set used is UTF-8. If a character is present in the UTF-8, then it is valid to Devanagari script otherwise not. After this, a document is divided into sentences along with word count. In Hindi sentences are separated using “danda viram (।)”. At the end of this step, a group of sentences is stored in a file and passed for further processing.

1.2. Tokenization –

Tokenization is the process of dividing the document into smaller parts i.e tokens. A token is a non-empty sequence of characters, excluding spaces and punctuation. Tokenization task is done by searching spaces between the words. The root words found are used for information retrieval applications.

1.3. Stop word removal-

Stop words are the most frequently occurring words which are generally used for sentence completion. These words are not important while considering the meaning of sentences. Hence we remove the stop words while scanning the document to fasten the speed of process by comparing them with a corpus of stop words.

1.4. Stemming-

The process of removing inflectional affixes (prefix and suffix) of words reducing the words to their stem is called stemming. It is done to get actual root word. We use suffix list to remove suffixes from words.

2. Feature extraction

We extract features from every sentence. All sentences are scored basis on the feature. We are using the following nine features to score the sentences.

2.1. Sentence Paragraph Position (f1)

Sentence position is important in the summarization of text document. Starting sentences of the paragraph have importance in the document as they convey the theme of a paragraph. Sentence position is calculated in a way that the starting sentence will have more importance.

Sentence paragraph position= (n-i)/n

 DOI : 10.18231/2454-9150.2018.1408

Where n is the total number of sentences in a paragraph and i is the position of the sentence in a paragraph.

2.2. Sentence Overall Position (f2)

Sentence position is calculated in the context of an entire document. Starting sentences have importance in a document as they convey the theme of a document. Sentence position is calculated in a way that the starting sentence will have more importance.

Sentence overall position= (n-i)/n

Where n is the total number of sentences in the document and i is the position of the sentence in the document.

2.3. Numerical Data in Sentence (f3)

Numeric data in text document represents important data such as date, rupees etc. Numeric data is calculated for every sentence such as

Numerical data value= (numeric data in sentence)/ (sentence length)

2.4. Presence of Inverted Commas (f4)

Important keywords in the document are presented in inverted commas. Presence of inverted commas in the sentence is calculated as

Presence of inverted commas= (Quoted words in the sentence)/ (Sentence length)

2.5. Sentence Length (f5)

Generally, short sentences do not carry important information. Also, sentences with too long length are not suitable for a summary. Sentence length is assigned such that medium length sentences will have more importance.

Sentence length value= (length of sentence) / (Length of longest sentence)

2.6. Keywords in Sentence (f6)

Keywords are the words which appear repeatedly in the document. Keyword value of the sentence is calculated as

Keyword value = (Total number of keywords)/ (Sentence length)

2.7. Title similarity (f7)

Sentences which contain words similar to the title are generally considered as important. Their value is calculated as

Title similarity value = (Total number of words similar to title)/ (Sentence length)

2.8. Term Frequency-Inverse Sentence Frequency (f8)

Term frequency calculates the distribution of word over the document. Inverse sentence frequency means the terms that occur in only a few sentences which are more important than others that occur in many sentences of the document. It is calculated as

tf-isf = tf*isf

Where tf = (total occurrence of the word in a sentence)/ (total words in the sentence)

And isf = log ((total sentences)/ (total sentences in which term occurred))

2.9. Sentence to sentence similarity (f9)

For every sentence of the document, the similarity between a sentence and rest other sentences of the document is calculated. By summing up all those similarity values, value sentence to sentence similarity is obtained.

3. Sentence ranking

At this stage ranks from 4-1 are given to the sentences based on features extracted. Each feature has assigned different weight. Based on this weight, the probability of sentence to be included in a summary is calculated. Probability is calculated using Naïve-Bayes rule. Later, the Naïve-Bayes classifier classifies the sentences in categories as a sentence to be included in summary and sentence not to be included in the summary [8]. The sentences extracted are included in the final summary file based on the total lines. First, all the sentences having rank 4 are included in the summary. Subsequently, sentences having rank 3 and 2 are included in the summary.

4. Generate summary

Finally, the summary of given input document is generated. It is given as final output to the user. Since it is an extractive summarization, generated summary will contain sentences from document itself. System will not modify the original structure of sentences.

Accuracy of the system will be tested with the manual extractive type of summary created by human experts.

V. CONCLUSION

In this paper, we have proposed the idea of creating an automatic extractive summarization system for summarizing the Hindi documents using Naïve-Bayes approach. Currently, the field of summarization is gaining importance as data on the internet is increasing at a very high rate. Instead of reading the entire document, reading the summary of a document is efficient. But creating a manual summary is a tedious task. Automatic summarization can provide summary at a very fast speed. Since all the features used in the feature extraction process are distinct, Naïve-Bayes provides simpler classification function. Such technique is mostly used for summarization English documents. Very few systems are available for Hindi documents.

In the future, this system can be tested with the addition of more features like cue words, context information, world

 DOI : 10.18231/2454-9150.2018.1408

knowledge etc. to improve the efficiency of summarization. Also, different supervised learning methods can be applied to improvise the performance.

REFERENCES

[1] C. Thaokar and L. Malik, "Test model for summarizing Hindi Text using extraction method", *Proceedings of 2013 IEEE Conference on Information and Communication Technologies*, 2013.

[2] D. Kaur and R. Kaur, "Automatic Summarization of Text Documents Written in Hindi Language", *International Journal of Computer Science and Mobile Computing*, vol. 3, no. 10, pp. 320-323, 2014. [Accessed 24 December 2018].

[3] V. Gupta and G. Lehal, "Automatic Text Summarization System for Punjabi Language", *Journal of Emerging Technologies in Web Intelligence*, vol. 5, no. 3, 2013. Available: 10.4304/jetwi.5.3.257-271.

[4] M. Ali, A. Al-Dahoud and B. Hawashin, "Enhanced Feature-Based Automatic Text Summarization System Using Supervised Technique", *INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY*, vol. 15, no. 5, pp. 6757-6767, 2016. Available: 10.24297/ijct.v15i5.1630.

[5] Bijal Dalwadi, et.al. "A Review: Text Categorization for Indian Language", 2349-4476, International Journal of Engineering Technology Management and Applied Sciences, March 2015.

[6] V. Dalal and D. Malik, "Automatic Summarization for Hindi Text Documents using Bio-inspired Computing", *IJARCCE*, vol. 6, no. 4, pp. 682-688, 2017. Available: 10.17148/ijarcce.2017.64130.

[7] S. Saziyabegum and P. S., "Literature Review on Extractive Text Summarization Approaches", *International Journal of Computer Applications*, vol. 156, no. 12, pp. 28-36, 2016. Available: 10.5120/ijca2016912574.

[8] Kupiec, J., Pedersen, J., & Chen, F. (1995, July). A trainable document summarizer. In Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 68-73). ACM.

[9] Prachi Shah and Nikita P. Desai, "A survey of automatic text summarization techniques for Indian and foreign languages", International conference on electrical, electronics and optimization techniques (ICEEOT) - 2016.

 DOI : 10.18231/2454-9150.2018.1408