MULTI DOCUMENT TEXT SUMMARIZATION USING BACKPROPAGATION NETWORK

Ashlesha Giradkar[1], S.D. Sawarkar[2], Archana Gulati[3]

[1] PG student, Datta Meghe College of Engineering
[2] Professor, Dept. of computer Engineering, DMCE , India
[3] Principal and Professor, Dept. of computer Engineering, DMCE , India

---

Abstract - *For English language lots of research work has been carried out in the field of text summarization but not for Hindi language. In the proposed system idea is to summarize multiple Hindi documents. This summarization is based on features extracted from documents such as sentence length, sentence position, sentence similarity, subject similarity etc. Thus, the proposed system can be used for Hindi text summarization of multiple documents based on backpropagation network.*

***Key Words*: Text summarization, Stemming, Hindi text summarization, Backpropagation Network, Sentence feature etc.**

1. INTRODUCTION

When we are talking about Text summarization, first we must be aware of what is a summary. Summary is a text that is produced from one or more texts documents that covers important information in the original text and it is shorter than original text document. The main aim of automatic text summarization is transform the source text into a shorter version which will further reduce reading time of user. Basically text Summarization methods can be classified into extractive and abstractive summarization. An extractive summarization method consists of selecting important sentences from the original document. An Abstractive summarization is an understanding the most ideas and concepts in a document and then rewriting it in own words. With the exponential growth in the quantity and complexity of information sources on the internet, it has become increasingly important to provide improved mechanisms to user to find exact information from available documents. The total system is work into three phases: pre-processing the text document, sentence scoring and summarization generation. This summarization is based on features extracted from documents such as sentence length, sentence position, sentence similarity etc. Thus, this system partially implemented for Hindi text summarization of multiple documents based on Backpropagation network.

2. EXPERIMENT

We use Hindi news articles, URLs as an input to summarization system. The text portion of Hindi news article fetched from URL is saved in a text document that acts as input documents to the summarizer. We used more than one news documents on same topic. Hindi news documents are collected from different news channels like Aajtak, dainikbhaskar etc.

3. APPROACH OF SUMMARIZATION

The proposed methods find out most relevant sentences from multiple Hindi documents by using statistical and linguistic approach. This summarization process has three major steps pre-processing, extraction of feature and implementation of backpropagation network.

3.1. Preprocessing

Preprocessing is nothing but preparing source document for analysis [8]. This preparation is basically going to perform in four steps sentence segmentation, sentence tokenization, stop word removal and stemming.

3.1.1. Segmentation

Given document is divided into sentences in segmentation step.

Ex. 1.भारत चीन सहयोग के तहत किए जा रहे इस युद्धाभ्यास में भारत के चुशूल गैरिसन और चीन के मोल्डो गैरिसन के सैनियों ने हिस्सा लिया

2. इससे पहले 6 फरवरी 2016 को इस तरह का अभ्यास किया गया था

3.1.2.Tokenization

In tokenization splitting of sentences into words takes place.
Ex. भारत, चीन, सहयोग, के,तहत, किए, जा, रहे, इस, युद्धाभ्यास, में, भारत, के, चुशूल, गैरिसन, और, चीन, के, मोल्डो, गैरिसन, के, सैनियों, ने, हिस्सा, लिया

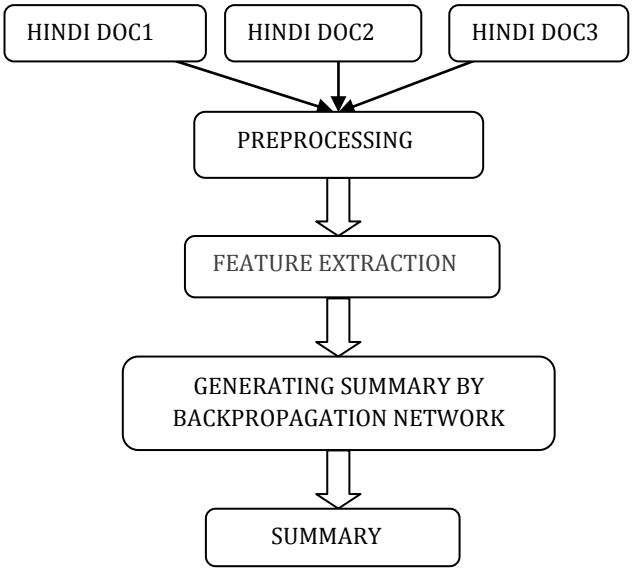

Fig-1: Proposed System

3.1.1. Segmentation

Given document is divided into sentences in segmentation step.

Ex. 1.भारत चीन सहयोग के तहत किए जा रहे इस युद्धाभ्यास में भारत के चुशूल गैरिसन और चीन के मोल्डो गैरिसन के सैनियों ने हिस्सा लिया

2. इससे पहले 6 फरवरी 2016 को इस तरह का अभ्यास किया गया था

3.1.2.Tokenization

In tokenization splitting of sentences into words takes place.

Ex. भारत, चीन, सहयोग, के,तहत, किए, जा, रहे, इस, युद्धाभ्यास, में, भारत, के, चुशूल, गैरिसन, और, चीन, के, मोल्डो, गैरिसन, के, सैनियों, ने, हिस्सा, लिया

3.1.3. Stop word removal

Sentence contains some word which do not aggregate relevant information for task are eliminated in this step.

Ex. के, किए, जा, रहे, इस, में, के, और, के, के, ने, लिया

3.1.4. Stemming

Stemming process find out root of word by removing prefix and suffix.

Ex. सैनिक is root word of सैनियों

3.2. Feature extraction

In feature extraction [11] step every sentence is represented by a vector of feature terms. Each sentence has a score based on the weight of feature terms which in turn is used for sentence ranking. Feature term values ranges between 0 to 1. Following section describes the features used in this study.

3.2.1. Average TF-ISF (Term Frequency Inverse Sentence Frequency)

TF means to evaluate distribution of each word over the document. "Inverse sentence frequency means that the terms that occurs in precisely some sentences that are additional necessary than others that occur in several sentences of the document." In other words, it is important to know in how many sentences a certain word exists. Since a word which is common in a sentence, but also it is common in the most of the sentences that is less useful when it comes to differentiating that sentence from other sentences.
This feature is calculated as "(1)"

$$TF = \frac{\text{Word occurrence in sentence (Si)}}{\text{Total number of words in (Si)}}$$

SF= Sentence frequency is count of sentence in which word occurred in a document of N sentences. So

IF = log [Total Sentences / SF]
tf*if = TF * IF

Average tf*if is calculated for each sentence that is nothing but weight of the sentence.

3.2.2. Sentence length

This feature is useful to filter out short or long sentences. Too short or long sentence is not good for summary. This feature computation uses minimum and maximum length threshold values. The feature weight is computed as "(2)".

SL = 0 if L <MinL or L >MaxL
Otherwise
SL = Sin ((L-MinL) * ((Max Θ- Min Θ)/ (MaxL - MinL)))
Where, L = Length of Sentence
MinL = Minimum Length of Sentence
MaxL = Maximum Length of Sentence
Min Θ = Minimum Angle (Minimum Angle=0)
Max Θ = Maximum Angle (Maximum Angle=180)

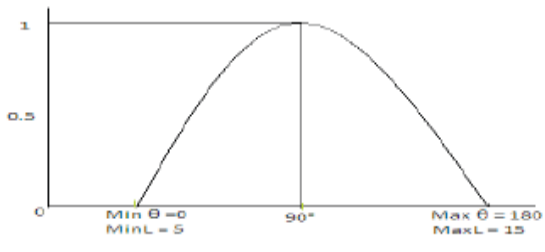

Fig-2: Sentence Length

3.2.3. Sentence position

Position of the sentence in the text, decides its importance. Sentences in the beginning defines the theme of the document whereas finish sentences conclude or summarize the document. In this threshold value in percentage defines how many sentences in the beginning and at the end are retained in summary whose weight is given as "3".

SP = 1.

Remaining sentences, weight is computed as follows

Sp = Cos ((CP - MinV)*((MaxΘ - MinΘ) / (MaxV - MinV)))
Where TRSH = Threshold Value
MinV = NS * TRSH (Minimum Value of Sentence)
MaxV = NS * (1 - TRSH) (Maximum Value of Sentence)
NS = Number of sentences in document
MinΘ = Minimum Angle (Minimum Angle=0)
MaxΘ = Maximum Angle (Maximum Angle=180)
CP = Current Position of sentence

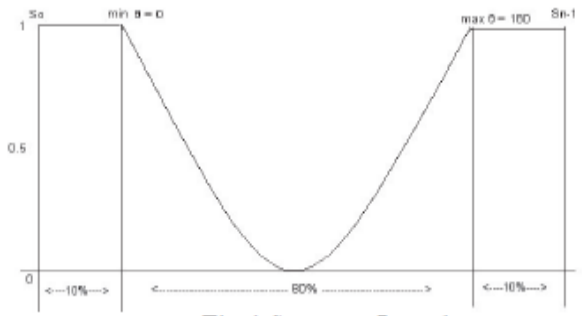

Fig-3: Sentence Position

3.2.4. Numerical data

The Sentence that contains numerical data is important and it should be included in the summary. The Weight for this feature is calculated as "4".
ND = 1, Digit exist
= 0, Digit does not exist

3.2.5. Sentence to sentence similarity

This feature finds the similarity between the sentences. For each sentence S, similarity between S and every other sentence is computed by the method of stemmed word matching.

$$Sim(l, m) = \frac{\text{Number of words occurred in Sentences (Sm)}}{WT}$$

Where, N = Number of Sentences
WT = Total Words in Sentence Si
Individual sentence weight based on similarity is the ratio of SS to N-1.

3.2.6. Title feature

Title contains set of words that represents gist of the document. So if a sentence Si has higher intersection with the title words then we can conclude Si is more important than other sentences in that document.

$$TS(Si) = \frac{\text{Number of words occurred in title}}{WT}$$

3.2.7. SOV qualification

Sentence is a group of words expressing a complete thought, and it must have a *subject* and a *verb*. The word order in Hindi is somewhat flexible. However, the typical word order of the most of the sentences is <subject><object><verb>. For this reason, Hindi is sometimes called an "SOV" language. For SOV qualification of every word of sentence are labeled as part of speech like (Noun, Adjective ,Verb, Adverb). The input to tagging algorithm is a set of words and specified tag to each. Tagging process is to look for the token in a look up dictionary. The dictionary used in this study is Hindi WordNet1.2 developed by IIT Mumbai. Word Net is an lexical database in which nouns, verbs, adjectives and adverbs are grouped organized into synonym sets or synsets, each representing one underlying lexical concept. A synset is a set of synonyms (word forms that relate to the same word meaning) and two words are said to be synonyms if their mutual substitution does not alter the truth value of a given sentence in which they occur, in a given context. Now based on the tags assigned, the first noun word in the sentence is marked as subject of the sentence. Whole sentence is parsed till its end, if verb is last word of the sentence than sentence is qualified as SOV. Only those sentence which are qualified as SOV will be used for further processing. Sentence after removing stop word is used.

SOV (Si) = 1, SOV Qualified
0, SOV Not Qualified

3.2.8. Subject similarity

For subject similarity feature, a result of previous step is employed to match subject of the sentence with the subject of the title. It can be similar to noun checking of title and sentence. Noun plays an important term in understanding the sentence. It is given as "8".
Sub (Si) = 1, if POS is noun and root value of title and sentence is equal
= 0, otherwise

3.3. Generating summary

In this step final generation of summary takes place. Summary generation is completed in two phases network training and sentence selection. Here network used is consists of simple three layer neuron structure input layer,

hidden layer and output layer. Input layer consist of 8 neurons. These neurons take input from 8 features as described above. Hidden layer consist of 300 neurons. Output layer consist of single neuron to indicate sentence should be in summary or not.

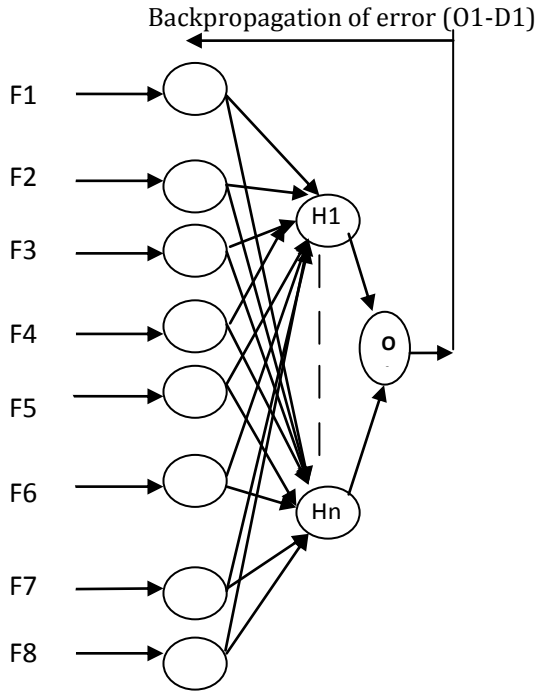

Fig-4: Network testing

3.3.1. Network training

The first phase of the process involves training the backpropagation network [9] to learn the types of sentences that should include in summary. This is done by training network with sentences from several text documents. With the help of input neurons, calculated value of features for sentences is fed to network. Further calculation of hidden layer and output layer are performed and this calculated output is compared with actual output of sentences. This procedure is repeated till this difference is below 0.01. During every cycle updation of weights takes place.

3.3.2. Sentence selection

Once the network has been trained, it can be used to filter sentences in paragraph and confirm whether or not sentence should be enclosed in the summary or not. As number of training data increases network performs better while selecting sentence that result in generation of more meaningful sentences in summary which give better idea about fed article.

4. EXPERIMENTS AND RESULTS

In this paper, the system is partially implemented & tested using Java. We used 70 Hindi news documents from different news based URLs as an input to the system. Here the human generated summaries are used as base summaries for evaluation of our results and on basis of base summary further calculation of backpropagation takes place. Since different user may be have different aspects for base summary, so there is flexibility in determining accuracy. Out of 70 document 50 document are used for training network and remaining 20 are used as testing document. It was observed that as number of training data increases system efficiency also increases.

5. CONCLUSIONS AND FUTURE SCOPE

In this work discussion is on Hindi text summarization using extractive method. An extractive summary is selection of important sentences from Hindi text documents. The importance of sentences is decided based on statistical and linguistic feature of sentences. This summarization procedure is based on back propagation network

This multiple Hindi text document summarizer will be useful for the summarization of various documents related to the same topic and come to a conclusion. This will be also helpful to get the summary of an important document in accurate and faster way. This project can be further enhanced to generate summary of multiple documents in various other Indian languages.

REFERENCES

1] Massih-Reza Amini, Patrick "Self-Supervised Learning for Automatic Text Summarization by Text-span Extraction" 23rd BCS European Annual Colloquium on Information Retrieval, 2001.

2] Khosrow Kaikhah" Automatic Text Summarization with Neural Networks "second IEEE international conference on intelligent systems, June 2004.

3] F. Kyoomarsi, h. Khosravi, e. Eslami and m. Davoudi "Extraction-based text summarization using Fuzzy analysis" Iranian Journal of Fuzzy Systems Vol. 7, No. 3, (2010) pp. 15-321.

4] Saeedeh Gholamrezazadeh, Mohsen Amini Salehi, Bahareh Gholamzadeh "A Comprehensive Survey on Text Summarization Systems" IEEE 2009.

5] Chetana Thaokar, Dr.Latesh Malik "Test Model for Summarizing Hindi Text using Extraction Method" Proceedings of 2013 IEEE Conference on Information and Communication Technologies (ICT 2013).

6] Nabil ALAMI, Mohammed Meknassi, Noureddine Rais "Automatic texts summarization: current state of the art" Journal of Asian Scientific Research, 2015.

7] Vishal Gupta "Hindi Rule Based Stemmer for Nouns "International Journal of Advanced Research in Computer Science and Software Engineering Volume 4, Issue 1, January 2014.

8] Mayuri Rastogi, Pooja Khana "Development of Morphological Analyzer for Hindi"

International Journal of Computer Applications (0975 – 8887) Volume 95– No.17, June 2014.

9] Dr. S. N. Sivanandam, Dr. S. N. Deepa "Principles of S.oft Computing"

10] Mr. Sarda A.T., Mrs. Kulkarni A.R. "Text Summarization using Neural Networks and Rhetorical Structure Theory" International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 6, June 2015

11] Arti S. Bhoir, Archana Gulati "Multi-document Hindi Text Summarization using Fuzzy Logic Method" International Journal of Advance Foundation And Research In Science & Engineering (IJAFRSE) Volume 2, Special Issue , Vivruti 2016.

12] Nitika Jhatta, Ashok Kumar Bathla "A Review paper on Text Summarization of Hindi Documents" IJRCAR VOL.3 ISSUE.5 may 2015