Bilingual Automatic Text Summarization Using Unsupervised Deep Learning

Shashi Pal Singh*1, Ajai Kumar*2, Abhilasha Mangal #1 , Shikha Singhal#2
*AAI, Center for development of Advanced Computing, Pune, India
*1shashis@cdac.in
*2 ajai@cdac.in
#Banasthali Vidyapith, Banasthali, Rajasthan, India
#1abhilasha9828@gmail.com
#2singhalshikha518@gmail.com

***Abstract:* In the world of digitization, the growth of big data is raising at large scale with usage of high performance computing. The huge data in English and Hindi is available on internet and social media which need to be extracted or summarized in user required form.**

In this paper we are presenting Bilingual (Hindi and English) unsupervised automatic text summarization using deep learning. which is an important research area with in Natural Language Processing, Machine Learning and data mining, to improve result accuracy, we are using restricted Boltzmann machine to generate a shorter version of original document without losing its important information. In this algorithm we are exploring the features to improve the relevance of sentences in the dataset.

***Keywords:* Automatic Summarization, Deep Learning RBM, Bilingual, dataset, unsupervised**.

I.INTRODUCTION

Text summarization or automatic text summarization corresponds to the process in which a computer creates a shorter version of the original text (or a collection of texts) still preserving most of the information present in the original text. This process can be seen as compression and it necessarily suffers from information loss [1].

Text summarization can be done on single document, which generates shorter version of it and multiple document summarizations, which generates summary from multiple related documents.

Text summarization is a method for data reduction. The use of text summarization enables users to reduce the amount of text that must be read while still assimilating the core information (Reeve et al., 2007). It helps user to find important and relevant information from large text document.

1.1 Motivation: Now a day's data on internet is gradually increasing. It is very difficult for human to remember complete data. A tool is needed that can minimize a large document into a smaller one while preserving its important information. Summarization is one such tool that helps to generate short form of original document which plays important role in information retrieval and information gathering. In this we are using Deep Learning approach which is emerging area in machine learning and helps to improve result accuracy. To enhance the feature values that we are extracting to generate summary here we are using Restricted Boltzmann Machine.

1.2 RBM (Restricted Boltzmann Machine): Restricted Boltzmann Machine is artificial neural networks which consist of visible layer and hidden layer and neurons of each layer has no connection between them but are connected to each neuron of other layer. Connections between layers are bidirectional and information flows in both direction. Layers of RBM form a bipartite graph.

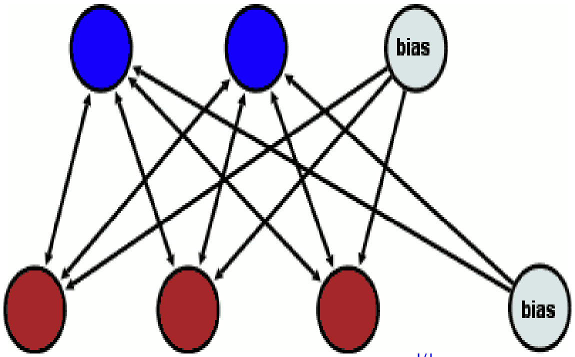


Fig. 1.: Restricted Boltzmann Machine[7]

1.3 Classification of Summarization: Automatic Summarization is process of reducing a text document with a computer program in order to create a summary

978-1-4673-9939-5/16/$31.00 ©2016 IEEE

that retains the most important points of the original document.

Summarization can be classified into two categories:

1.3.1 Extractive Text Summarization – In this method subset of words, phrases and sentences are selected from original text to generate a summary.

1.3.2 Abstractive Text Summarization- Abstractive method uses natural language generation technique to create a summary using internal semantic representation.

1.4 Applications of Automatic Summarization:

1.4.1 Automatic Summarization can be used for natural language processing task such as question answering, Text Classification, or Information retrieval.

1.4.2 It can be used to present compressed description of search result in search engine.

1.4.3 To summarize news for mobile phones/PDA.

1.4.4 Text Summarization can also be useful for text display on hand-held devices, such as PDA. For instance, summarized version of an email can be sent to a hand-held device instead of a full email [1].

II. RELATED WORK

Already Implemented System

MEAD: MEAD[14] is the most elaborate publicly available platform for multi-lingual summarization and evaluation. The platform implements multiple summarization algorithms such as position-based, centroid-based, largest common subsequence, and keywords. The methods for evaluating the quality of the summaries are both intrinsic and extrinsic. MEAD implements a battery of summarization algorithms, including baselines (lead-based and random) as well as centroid-based and query-based methods.

- Neural Network is used by S.P yong[9]. He used keywords extraction and summary production system to generate summary.
- RST is used by Li Chengcheng[10] to analyze sentence and discover rhetoric relations to generate a Summary.
- In 2000 Hongyan Jing[11] take closely related sentences for this he used human abstraction concept.
- In 2011 Nitin Agarwal[12] used unsupervised query-oriented approach with the help of clustering based method.
- In 2004 Jun'ichi Fukumoto[13] using TF/IDF for single and multiple documents abstract generation.

III. PROPOSED APPROACH

Automatic Summarization is done by Extractive and Abstractive method both but natural language generation has certain limitation due to which Abstractive text Summarization is difficult so generally Extractive Text Summarization is done. Proposed Approach can be classified into following phases Pre-processing phase, Feature extraction phase, Deep Learning Algorithm and Post processing phase. Our Approach works well for both English and Hindi language. In Text Summarization, first Pre-processing of text is done and then Sentence Feature matrix is generated which is further used to calculate sentence scores and according to those scores summary of that text is generated.

Flow Chart of Proposed approach is as follows:

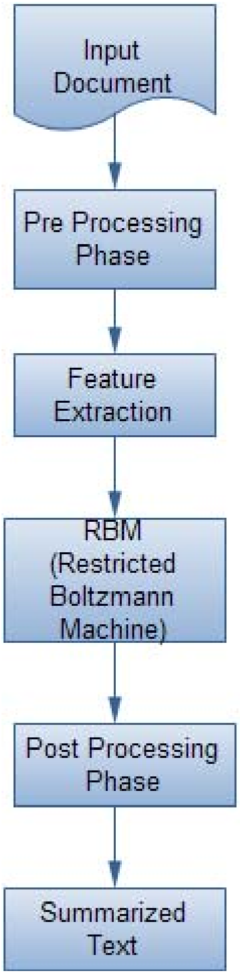


Fig. 2. Flow Chart of Text Summarization

Process to generate summary is as follows:

A. *Input Phase:* User can input a file, a Web URL or write text. If user enters a file or web URL then text is extracted from text extraction module

B. *Pre Processing Phase:* In this phase, Text is pre-processed to remove the unwanted things from text. Following techniques are used for Pre Processing:

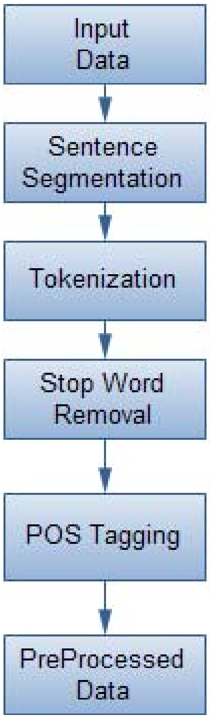


Fig. 3 Pre Processing Flow Chart

- *Sentence Segmentation:* We have divided the complete text into sentences. We have done English Sentence extraction using Open NLP library (Apache Software Foundation), which detects sentences from document, and Hindi Sentence extraction is done on basis of Hindi rules for sentences.
- *Tokenization:* Document is tokenized to generate tokens that are used to detect Keywords and Key phrases and calculating each Term frequency in document.

- *Stop Word Removal:* Stop words are high frequency words of a language that do not carry particular information of their own, which are filtered out before or after processing of natural language data. Any group of words can be chosen as stop words. We have chosen “is, am, are, the, he, she” etc as stop words.

- *Part Of Speech Tagging:* In part-of speech-tagging, Words are tagged on basis of its category (noun, verb, adverb, adjective) and its context. In this we are using RDR POS Tagger (Dat Quoc Nguyen, 2013-2015) to do part of speech tagging.

C. *Feature Vector Calculation:* After pre processing of text, each sentence is represented by feature vector. Various Features are calculated on each sentence to generate a feature matrix which is further used to calculate sentence score. We have considered following features:

- *Sentence Position Feature:* Sentence relevance can be checked on the basis of its position in document. We have calculated position feature as follows:

  Sentence_Position = 1, if sentence is the first or last sentence of text.
  Sentence_Position = cos((sentencePos-minv)*((1/maxv)-minv)), for others.
  Where,
  sentencePos is position of sentence in the text.
  minv is calculated as (th*N)
  maxv is calculated as (th*2*N)
  N is total number of sentences in document.
  th is threshold calculated as (0.2*N)

- *Sentence TF-ISF Feature:* In information retrieval Tf-Idf(Term frequency * Inverse document frequency) is very useful feature. We can employ that feature into text summarization. Here processing is on single document so we have taken Tf-Isf (Term frequency * Inverse Sentence Frequency) feature into account in which term frequency of each term in particular sentence is multiplied by Isf which is calculated as total number of occurrences of the term in all the other sentences.

  Tf-Isf=(log (isf)*(tf))/length

  Where, isf is total number of occurrences of the each term of ith sentence in all the other sentences and tf is term frequency of each term in ith sentence.

  length is total number of words in sentence$_i$

- *Sentence to Centroid Similarity Feature:* In sentence to centroid feature one sentence from whole document is considered as centroid. We calculated the Centroid sentence on basis of Tf-Isf feature. Sentence whose Tf-Isf feature is maximum is considered as centroid sentence. Then cosine similarity of each sentence is calculated with that sentence centroid to calculate this feature value.

  Centroid_feature$_i$=cos (sentence$_i$, centroid)

  Where, 1<=i<=N,
  N is total No. of sentences in document,
  Centroid_feature$_i$ is centroid feature of ith sentence
  sentence$_i$ is ith sentence of document.

- *Sentence to Sentence Similarity Feature:* Sentence to Sentence similarity feature is calculated as follows, for each sentence (S) similarity is calculated with each other sentence(S’) in document then all those similarity values are added up to generate sentence similarity feature, this process is repeated for all the sentences in document.

SentenceSimilar_feature$_i$= $\sum_1^N j\ similar$ (sentence$_i$, sentence$_j$)

Where, 1<=i<=N
SentenceSimilar_feature$_i$ is feature value of sentence i. and Sentence$_i$ is ith sentence of document.

- *Numeric Token Feature:* Numeric token feature is calculated as follows, In each sentence(S) we calculated total number of numeric words in S divided by total number of words in sentence.

  Numerictoken_feature$_i$ = num_numeric$_i$ /length

  Where, num_numeric is number of numeric token in sentence i.
  Length is total number of words in sentence

- *Sentence Length Feature:* Sentence Length feature is used to exclude too short sentences in summary as too short sentences are not important for summary.

  Sentencelength_feature$_i$ = 0
  (if number of words in sentencei is less than 3)
  Sentencelength_featurei =number of words in sentence I and Where, sentencei is ith sentence of document

- *Proper Noun Feature:* Proper nouns are words used to classify class of people, place, or thing. Proper Noun Feature is taken into account to calculate total number of proper nouns in each sentence i. Proper Nouns are calculated on basis of part-of speech tagging of each sentences. This feature is used to give importance to those sentences, which has proper nouns.

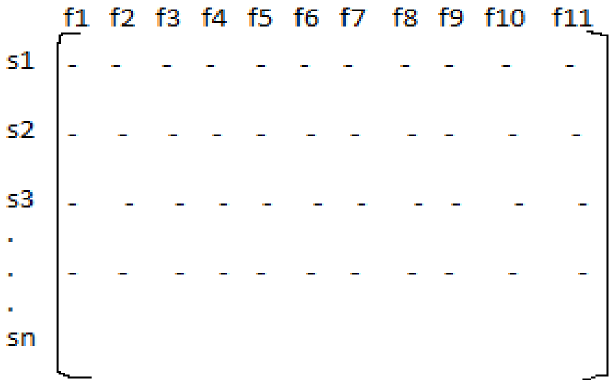
- *Named Entity Feature:* In this feature, we are calculating total number of named entities in each sentence(s). Occurrence of named entity in sentence indicates the relevance of sentence.

- *Unique Term Feature:* This feature detects whether or not a sentence contains Unique keywords of documents. Unique terms are calculated by tokenizing the complete document and after removing the stop words. This feature represent whether or not the sentence captures the main concept of document or not. Unique term feature for sentence$_i$ is total number of unique term in that sentence.

- *Bi Gram Key Phrase Feature*: Bi Gram Key feature calculates total no of relevant bi gram key phrases in ith sentence. Bi key phrases are generated for each sentence taking bi grams and if any word of bi gram is a stop word then it is removed from bi gram key phrases list and only one with no stop word is taken into consideration and added to key phrases list . This feature is useful to check sentence relevance on basis of total no of bi key phrases in that sentence.

- *Tri Gram Key Phrase Feature:* Tri Gram Key feature calculates total no of relevant Tri gram key phrases in ith sentence. Tri key phrases are generated for each sentence taking Tri grams and if any word of Tri gram is a stop word then it is removed from Tri gram key phrases list and only one with no stop word is taken into consideration and added to key phrases list. This feature is useful to check sentence whether sentence has key phrase or not.

D. *Sentence Matrix Generation:* After calculating the feature values of each sentence of document then we have generated a sentence matrix on basis of feature values calculated for each sentence. Sentence Matrix is a 2-d matrix here Sentence matrix S=(s_1 , s_2 , s_3 , ….. , s_N) where s_i=(f_1 , f_2 , f_3 , f_4 , f_5 , f_6 , f_7 , f_8 , f_9 , f_{10} , f_{11}) is a feature vector and i<=N. where N is total number of sentences in document.



$$\begin{array}{c|ccccccccccc|}
 & f1 & f2 & f3 & f4 & f5 & f6 & f7 & f8 & f9 & f10 & f11 \\
s1 & - & - & - & - & - & - & - & - & - & - & - \\
s2 & - & - & - & - & - & - & - & - & - & - & - \\
s3 & - & - & - & - & - & - & - & - & - & - & - \\
. & & & & & & & & & & & \\
. & - & - & - & - & - & - & - & - & - & - & - \\
. & & & & & & & & & & & \\
sn & & & & & & & & & & &
\end{array}$$

Fig. 4. Feature matrix For Text Summarization

E. *Deep Learning Approach:* The sentence matrix S is generated where each s_i contains all eleven-feature vector values. After this, the recalculation is done on matrix S to enhance the feature vector, so that more accurate summary can be generated.

To enhance, the feature matrix S is given as input to RBM(Restricted Boltzmann Machine) which has two hidden layers each sentence is first passed through hidden layer 1 In which feature values of each sentence is multiplied by randomly generated weights and one bias value is randomly generated which is added to all the sentence. The same procedure is repeated for hidden layer2 where output of hidden layer1 is given as input to hidden layer2.

Two bias vectors are taken such as:

$$B_0 = [b_1 , b_2 , \dots , b_n]$$
$$B_1 = [b_1 , b_2 , \dots , b_n]$$

To get the more refined set of sentence features, RBM works in two steps.

- The input to first step is sentence matrix, S = (s1,s2,……..sn), which is having the eleven features of sentence as element of each sentence set. During the first cycle of RBM, a new refined sentence matrix set:

$$S=(s_i' ,s_2' ,\dots\dots,s_n')$$

Where, 1<=i<=n and s_i' is calculated as

$$s_i'=s_i*w_i +b_i$$

Where, w_i is randomly generated weight value and b_i is bias value from B_0

- Same operation is repeated again at hidden level 2 where input to this step is output of hidden layer1, $S= (s_1' , s_2' ,\dots\dots, s_n')$ and output is more enhanced feature value.

$$S=(s_1'' , s_2'' , \dots , s_n'').$$

6.Sentence Score: After calculating the enhanced Sentence matrix we calculated the sentence score of each sentence of document. Sentence score is calculated by adding all the enhanced feature values of that sentence**.**

$$\text{Sentence\_score}_i = \sum_1^{11} j\ (f_j)$$

Where, Sentence_score_i is sectence score of i^{th} sentence and f_j is j^{th} feature value of sentence i.

7. Summary Generation: All the sentences are sorted in descending order on basis of their sentence score. In this method, we have considered first sentence of summary as most important sentence. First sentence is always included in summary then top 50% sentences from sorted set are taken and cosine similarity of first sentence is calculated with those sentences.

The sentence with highest similarity is added to list and then cosine similarity of that added sentence is calculated with top 50% sentences. Same procedure continues till we get total number of sentences that we want in summary. Summary limit is taken from user and by default it is set to 10sentences. After that, we have arranged those sentences based on their sentence position in original text. These arranged sentences represent the summary of the text.

IV. EXPERIMENTS AND RESULTS

We are using evaluation toolkit ROUGE[15] (Recall-Oriented Understudy for Gisting Evaluation) which is a software package for evaluation of automatic summarization in natural language processing.

It compares human generated summary and the system generated summary. We are using ROUGE-1 as it has high recall significance test. TABLE-1 represents the recall, precision and F-measure of Hindi and English documents calculated taking human generated summary as one reference and our system-generated summary as another reference. According to F-Measure our system is giving 85% accuracy.

Table 1. Recall, Precision and F-Measure among Documents

Document	Recall	Precision	F-Measure
EngDoc1	0.87097	0.8813	0.87610455
HINDoc1	0.7469	0.7912	0.76841204
HINDoc2	0.76333	0.8162	0.78888016
HINDoc3	0.83669	0.98163	0.90338334
EngDoc2	0.71667	0.7418	0.7290185
EngDoc3	0.875	0.884	0.87947697
EngDoc4	0.88848	0.9042	0.89627107
HINDoc4	0.81818	0.87907	0.84753276
EngDoc5	0.98171	0.97281	0.97723973
Average	0.83310333	0.87246777	0.85233129

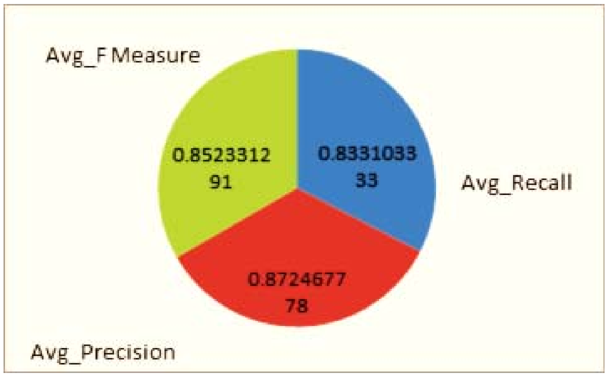


Fig. 5. Average Recall, Precision and F-Measure Pie Chart

Fig.5. represents the pie chart of average Recall, Precision and F-Measure Score which shows average recall is 83%, Precision is 87% and F-Measure is 85%.

V. CONCLUSION AND FUTURE WORK

Much Research has been done in field of text summarization and most of them are using supervised approach. We have developed an automatic summarizer, which works on two languages Hindi and English using unsupervised deep learning approach. Here we are extracting eleven features from each sentence of document and generating the feature matrix.

The generated feature matrix is then passed through Restricted Boltzmann Machine to enhance importance of relevant sentences. Open source technologies are used to implement the proposed algorithm. The output result of proposed algorithm is almost 85% accurate and also preserves the meaning of summarized document. In future, enhancement can be done by adding more features to get more relevant sentences and meaningful summary and further we will be applying the concept to generate multiple documents summarization.

REFERENCES

[1] Naresh Kumar Nagwani, Dr. Shrish Verma,"A Frequent Term and Semantic Similarity based Single Document Text Summarization Algorithm", International Journal of Computer Applications (0975 – 8887) Volume 17– No.2, March 2011.

[2] Reeve, LH, H Han and AD Brooks (2007). “The use of domain-specific concepts in biomedical text summarization”, Information Processing & Management, 43(6), 1765–1776.

[3] https://opennlp.apache.org, (Accessed: 16 November,2015).

[4] RDRPOSTagger , Rule-based Part-of-Speech and Morphological Tagging Toolkit is implemented by Dat Quoc Nguyen, Dai Quoc Nguyen, Dang Duc Pham, and Son Bao Pham.

[5] Neelima Bhatia,Arunima Jaiswal,"Literature Review on Automatic Text Summarization: Single and Multiple Summarizations", International Journal of Computer Applications (0975 – 8887)Volume 117 – No. 6, May 2015.

[6] Joel Larocca Neto Alex A. Freitas Celso A. A. Kaestner,"Automatic Text Summarization using a Machine Learning Approach".

[7] PadmaPriya, G. and K. Duraiswamy, "An Approach for text summarization using deep learning algorithm", Journal of Computer Science 10 (1): 1-9, 2014.

[8] J. Anitha, P. V. G. D. Prasad Reddy and M. S. Prasad Babu, “An Approach for Summarizing Hindi Text Through a Hybrid Fuzzy Neural Network Algorithm”, Journal of Information & Knowledge Management, Vol. 13, No. 4 (2014).

[9] S. P. Yong, A. I. Z. Abidin and Y. Y. Chen, “A Neural Based Text Summarization System”, 6th International Conference of Data Mining, pp. 45-50, 2005.

[10] LiChengcheng, “Automatic Text Summarization Based On Rhetorical Structure Theory”, International Conference on Computer Application and System Modeling (ICCASM), vol. 13, pp. 595-598, October 2010.

[11] Hongyan Jing, “Sentence Reduction for Automatic Text Summarization”, In Proceedings of the 6th Applied Natural Language Processing Conference, Seattle, USA, pp. 310-315, 2000.

[12] Nitin Agarwal, Gvr Kiran, Ravi Shankar Reddy and Carolyn Penstein Ros´e, “Towards Multi-Document Summarization of Scientific Articles: Making Interesting Comparisons with SciSumm”, Proceedings of the Workshop on Automatic Summarization for Different Genres, Media, and Languages, Portland, Oregon, pp. 8–15.

[13] Jun'ichi Fukumoto, “Multi-Document Summarization Using Document Set Type Classification”, Proceedings of NTCIR- 4, Tokyo, pp. 412-416, 2004.

[14] http://www.summarization.com/mead,(Accessed:20 November,2015).

[15] C.Y. Lin, “ROUGE: A Package for Automatic Evaluation of Summaries” In Proceedings of Workshop on Text Summarization of ACL, .Spain. 2004.