Journal of Computer Science 10 (1): 1-9, 2014
ISSN: 1549-3636

doi:10.3844/jcssp.2014.1.9 Published Online 10 (1) 2014 (http://www.thescipub.com/jcs.toc)

AN APPROACH FOR TEXT SUMMARIZATION USING DEEP LEARNING ALGORITHM

[1]PadmaPriya, G. and [2]K. Duraiswamy

[1]Department of Computer Science and Engineering, K.S.R. College of Engineering,
[2]Department of Computer Science and Engineering, K.S. Rangasamy College of Technology,
K.S.R. Kalvi Nagar, Tiruchengode, Tamilnadu, India

Received 2012-09-17, Revised 2013-07-06; Accepted 2013-10-07

ABSTRACT

Now days many research is going on for text summarization. Because of increasing information in the internet, these kind of research are gaining more and more attention among the researchers. Extractive text summarization generates a brief summary by extracting proper set of sentences from a document or multiple documents by deep learning. The whole concept is to reduce or minimize the important information present in the documents. The procedure is manipulated by Restricted Boltzmann Machine (RBM) algorithm for better efficiency by removing redundant sentences. The restricted Boltzmann machine is a graphical model for binary random variables. It consist of three layers input, hidden and output layer. The input data uniformly distributed in the hidden layer for operation. The experimentation is carried out and the summary is generated for three different document set from different knowledge domain. The f-measure value is the identifier to the performance of the proposed text summarization method. The top responses of the three different knowledge domain in accordance with the f-measure are 0.85, 1.42 and 1.97 respectively for the three document set.

Keywords: Multi-Document, Summary, Redundancy, RBM, DUC 2002 Dataset (Document Understanding Conferences)

1. INTRODUCTION

From many years, summarization is done by humans manually. In the present time, the amount of information is increasing gradually by the mean of internet and by other sources. To overcome this problem, text summarization is essential to tackle the overloading of information. Text summarization helps to maintain the text data by following some rules and regulations for efficient usage of text data. For example, the extraction of summary from a given document for the extraction of a definite content from the whole document or multi-documents. Text summarization relates to the process of obtaining a textual document, obtaining content from it and providing the necessary content to the user in a shortened form and in a receptive way to the requirement of user or application. Automatic summarization is linked closely with text understanding which imposes several challenges comprising of variations in text formats, expressions and editions which adds up to the ambiguities (Sharef *et al*., 2013). Researchers in text summarization have approached this problem from many aspects such as natural language processing (Zhang *et al*., 2011), statistical (Darling and Song, 2011) and machine learning and text analysis is the fundamental issue to identify the focus of the texts.

Text summarization can be classified in two ways, as abstractive summarization and extractive summarization. Natural Language Processing (NLP) technique is used for parsing, reduction of words and to generate text summery inabstractive summarization. Now at present NLP is a low cost technique and lacks in precision. Extractive summarization is flexible and consumes less time as compared to abstractive summarization (Patil and Brazdil, 2007). In extractive summarization it consider all the sentence in a matrix form and on the basis of some

Corresponding Author: PadmaPriya, G., Department of Computer Science and Engineering, K.S.R. College of Engineering, K.S.R. Kalvi Nagar, Tiruchengode, Tamilnadu, India

feature vectors all the necessary or important sentences are extracted. Afeature vector is an n-dimensional vector of numerical features that represent some object. The main objective of text summarization based on extraction approach is the choosing of appropriate sentence as per the requirement of a user.

Generally, text summarization is the process of reducing a given text content into a shorter version by keeping its main content intact and thus conveying the actual desired meaning (Mani, 2001a; 2001b). Single document summarization is a process, which deals with a single document only. Multi-document summarization is the method of shortening, not just a single document, but a collection of related documents, into a single summary (Ou *et al*., 2008). The concept looks easy, but while implementation it is a tough task to compile. Sometimes it may not be able to fulfill our desired goal. Most of the similar techniques employed in single-document summarization are also employed in multi-document summarization. There exist some notable disparities (Goldstein *et al*., 2000): (1) The degree of redundancy contained in a group of topically-related articles is considerably greater than the redundancy degree within an article, since each article is appropriate to illustrate the most important point and also the required shared background. So, anti-redundancy methods play a vital role. (2) The compression ratio (that is the summary size with regard to the size of the document set) will considerably be lesser for a vast collection topically related documents than for single document summaries. In order to provide a lot of semantic information, guided summarization task is introduced by the Text Analysis Conference (TAC). It aims to produce semantic summary by using a list of important aspects. The list of aspects defines what counts as important information but the summary also includes other facts which are considered as especially important. Furthermore, an update summary is additionally created from a collection of later Newswire articles for the topic under the hypothesis that the user has already read the previous articles. The summary generated is guided by pre-defined aspects that is employed to enhance the quality and readability of the resulting summary (Kogilavani and Balasubramanie, 2012).

In this study, we have developed a multi-document summarization system using deep learning algorithm Restricted Boltzmann Machine (RBM). Restricted Boltzmann Machine is an advance algorithm based on neural network, it performs the entire necessary task for text summarization. Initially, the preprocessing steps are applied, those steps include (1) Part of speech tagging, (2) Stop word filtering, (3) steaming. Then comes the feature extraction part. In this part of the text summarization certain features of sentences are extracted. The features we are extracting are: Title Similarity, Positional Feature, Term Weight and Concept Feature. All most all the text summarization models face two major problems, first the ranking problem and the second one is how to create the subset of those ranking or top ranked sentences. There are varieties of approaches for the ranking problem. In this study we are solving the ranking problem by finding out the intersection between the user query and a particular sentence. On the basis of this, a sentence score is generated for every sentence and they are arranged in descending order. Out of this ranked sentences some of sentences are selected on the basis of compression rate entered by the user. In this way we solve the ranking problem. In the end we have used DUC 2002 dataset to evaluate the summarized results based on the measures such as Precision, recall and f-measure.

1.1. Motivation

Now days more and more information is available through internet and other sources. To handle these data more efficiently we need a tool for extracting proper set of sentences from the given documents. Summarization of text is essential to get the important information while dealing with large collection of documents. With the advent of World Wide Web information has become intrinsic part of our life. To remember the details of every information is not possible for human mind. Therefore summarization of text documents plays a very important role in information gathering. In this study we are using deep learning Algorithm for the summarization task. Deep learning is the emerging field of machine learning, which is used to solve problems of number of computer science domain like image processing, robotics, motion. Recently it is also used in domain of Natural language processing with very encouraging results. An algorithm is deep if its input is passed through several of nonlinearity's before being output most modern learning algorithms includingsvm and naive ayes classifier are shallow. Here we are using the Restricted Boltzman Machine to extract the top most feature word of text.

1.2. Restricted Boltzman Machine

Restricted Boltzmann Machine is a stochastic neural network (that is a network of neurons where each neuron has some random behavior when activated).

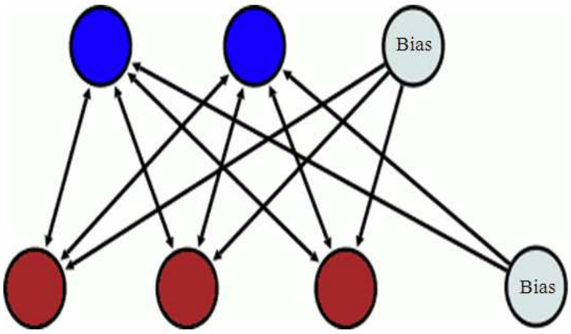

Fig. 1. Restricted boltzmann machine

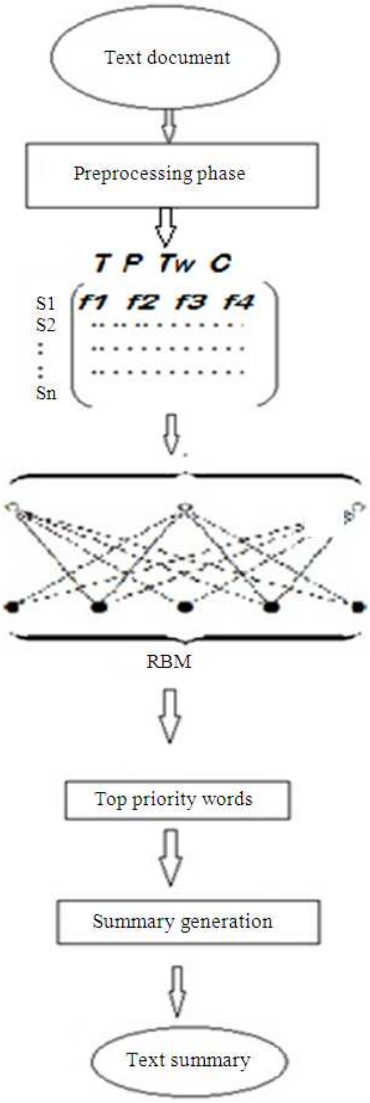

Fig. 2. Block diagram of text summarization

It consist of one layer of visible units (neurons) and one layer of hidden units. Units in each layer have no connections between them and are connected to all other units in other layer (**Fig. 1**). Connections between neurons are bidirectional and symmetric. This means that information flows in both directions during the training and during the usage of the network and those weights are the same in both directions.

1.3. RBM Network Works in the Following Way

First the network is trained by using some data set and setting the neurons on visible layer to match data points in this data set.

After the network is trained we can use it on new unknown data to make classification of the data (this is known as unsupervised learning).

1.4. Proposed Deep Learning Approach

Text summarization technique is divided into two approaches extractive and abstractive. But due to the limitation of natural language generation techniques in generating the abstractive summary generally extractive approach is used for summarization. For summarizing the text there is a need of structuring the text into certain model which can be given to RBM as input. First of all in text summarization the text document is preprocessed using various prevalent preprocessing techniques and then it is converted into sentence matrix defined over a vocabulary of words. This structured matrix each row will work as a input to our RBM (**Fig. 2**). After getting the set of top priority word from the RBM the input query, sentence vector and high priority word output is compared to generate the extractive summary of the text document.

1.5. Preprocessing

To make the document light (not containing unwanted words) preprocessing of the text document for structuring is done by applying various techniques developed by the linguist. There are myriads of technique by which we can reduce the density of text document. In this study we are using the following techniques.

1.6. Part of Speech Tagging

Part of speech tagging is the process of marking or classifying the words of text on the basis of part of speech category (noun, verbs, adverb, adjectives) they belong. Varieties of algorithms are there to perform the POS tagging like hidden Markova models, using dynamic programming.

1.7. Stop Word Filtering

Stop words are the words which are filtered out prior to or after the preprocessing task generally there is no specific rule on aparticular word to be stop word, it is completely subjective depends upon the situation. In our condition we considering words like a, an, in by as stop word and filters this word from the original document. Stop word filtering is the standard filtering in text mining applications.

1.8. Stemming

Another important technique we need to apply is steeming. Steeming is process of bringing the word to its base or root form for example using words singular form instead of using the plural (using boys as boy), removing the ing from verb (changing doing to do). There are number of algorithms, generally referred as stemmers', are there that can be used to perform the stemming.

1.9. Feature Vector Extraction

After reducing the density of document, the document is structured into a matrix. A sentence matrix S of order n*v is containing the features for every sentence of a matrix. For very informative summarization we are extracting four features of a sentence of text document viz similarity with title, relative position of sentence, term weight of words forming sentences, concept-extraction of sentence. Sentence matrix row vector represents the sentence which is making the document and column vector contains the entry for these extracted features.

1.10. Feature Computation

1.10.1. Title Similarity

A sentence is considered important if it's similar to the title of text document. Here similarity is considered on the basis of occurrence of common words in title and sentence. A sentence has good feature score if it has maximum number of words common to the title. The ratio of the number of words in the sentence that occur in title to the total number of words in the title helps to calculate the score of a sentence for this feature. It is calculated by:

$$f1 = \frac{s \cap t}{t}$$

Where:

S = Set of words of sentence
T = Set of words of title
$s \cap t$ = Common words in sentence and title of document

1.11. Positional Feature

Positional value of a sentence is also extracted. A sentence is relevant or not can also be judged by its position in the text. To calculate the positional score of sentence we are considering the following conditions:

f2 = 1, if sentence is the starting sentence of the text
f2 = 0, if sentence comes in the middle paragraphs of text
f2 = 1, if sentence comes in the last of the text

1.12. Term Weight

This is another very important feature to be consider for summarization of text. Here by term weight we simply mean the term frequency and its importance. This is the most standard feature considered in various natural language processing tasks. The frequency here is the term frequency which reflects the importance of a word in a document, it simply tells number of times a word appears in the text. The term frequency of a word will be given by tf(f,d) where f is the frequency of the word and d is text the document. The total term weight is calculated by computing tf(f,d) and idf for a document. Here idf refers to inverse document frequency which simply tells about whether the term is common or rare across all documents. It is obtained by dividing the total number of documents by the number of documents containing the term and then taking the log of that quotient. The idf is given by:

$$\mathrm{idf}\left(t,D\right) = \log \frac{D}{d \in D : t \in d}$$

where, D is the total number of documents, $\in$D: t$\in$d, it is the number of documents where term t appears. The total term weight is given by tf*idf which can be calculated by:

$$\mathrm{tf} * \mathrm{idf}\left(t,d,D\right) = \mathrm{tf}\left(t,d\right) * \mathrm{Idf}(t,D$$
$$f3 = \mathrm{tf} * \mathrm{idf}.$$

1.13. Concept Feature

The concept feature from the text document is extracted using the mutual information and windowing process. In windowing process a virtual window of size 'k' is moved over document from left to right. Here we want to find out the co-occurrence of words in same window and it can be calculated by following formula:

$$MI(w_i, w_j) = \log 2 \frac{P(w_i, w_j)}{P(w_i) * P(w_j)}$$

where, $P(w_i, w_j)$-joint probability that both keyword appeared together in a text window.

$P(w_i)$-probability that a keyword w_i appears in a text window and can be computed by:

$$P(w_i) = \frac{|sw_t|}{|sw|}$$

Where:

sw_i = The number of windows containing the keyword w_i

$|sw|$ = Total number of windows constructed from a text document

The sentence matrix generate by above steps is:

$$\begin{array}{c} S1 \\ S2 \\ . \\ . \\ Sn \end{array} \begin{pmatrix} T & P & Tw & C \\ f1 & f2 & f3 & f4 \\ .. & ... & .. & .. \\ .. & ... & .. & .. \\ ... & .. & .. & .. \end{pmatrix}$$

1.14. Sentence Matrix

Here sentence matrix S = $(s_1, s_2, \ldots\ldots s_n)$ where $s_i = (f_1, f_2, \ldots\ldots f_4)$, i<= n is the feature vector.

1.15. Deep Learning Algorithm

The sentence matrix S = $(s_1, s_2, \ldots\ldots s_n)$ which is the feature vector set having element as s_i which is set contains the all the four features extracted for the sentence s_i. Here this set of feature vectors S will be given as input to deep architecture of RBM as visible layer. Some random values is selected as bias H_i where i = 1,2 since a RBM can have at least two hidden layer. The whole process can be given by following equation:

$$S = (s_1, s_2 \ldots\ldots s_n)$$

where, $s_i = (f_1, f_2, \ldots\ldots f_4)$, i<= n where n is the number of sentences in the document. Restricted Boltzmann machine contains two hidden layers and for them two set of bias value is selected namely H_0H_1:

$$H_0 = \{h_0, h_1, h_2 \ldots\ldots h_n\}$$
$$H_1 = \{h_0, h_1, h_2 \ldots\ldots h_n\}$$

These set of bias values are values which are randomly selected. The whole operation of Sentence matrix is performed with these two set of randomly selected value. The whole operation with RBM starts with giving the sentence matrix as input. Here $s_1, s_2, \ldots\ldots s_n$ are given as input to RBM. The RBM generally have two hidden layers as we mentioned above.

Two layers are sufficient for our kind of problem. To get the more refined set of sentence features. RBM works in two step. The input to first step is our set of sentence matrix, S = $(s_1, s_2, \ldots\ldots s_n)$, which is having the four features of sentence as element of each sentence set. During the first cycle of RBM a new refined sentence matrix set:

$$s' = (s'_1, s'_2, \ldots\ldots s'_n)$$

The above expressed s'is generated by performing:

$$\sum_1^n s_i + h_i$$

During step 2 the same procedure will be applied to this obtained refined set to get the more refined sentence matrix set with H_1 and which is given by:

$$s'' = (s''_1, s''_2, \ldots\ldots s''_n)$$

After obtaining the refined sentence matrix from the RBM it is further tested on a particular randomly generated threshold value for each feature we have calculated. For example we select threshold thr_c as a threshold value for the extracted concept-feature. If for any sentence f_4<thr then it will be filtered and will become member of new set of feature vector.

Step 1. $s_1, s_2 \cdots, s_n$

$$[f1, f2, f3, f4] \quad [f1, f2, f3, f4] \quad [f1, f2, f3, f4]$$
$$\searrow \qquad \downarrow \qquad \swarrow$$
$$\sum_1^n s_i + h_i(H_0)$$
$$\downarrow$$
$$s' = (s'_1, s'_2 \cdots, s'_n)$$

Step 2. $s'_1, s'_2 \cdots, s'_n$

$$\begin{tikzcd}
{[f1,f2,f3,f4]} \arrow[dr] & {[f1,f2,f3,f4]} \arrow[d] & {[f1,f2,f3,f4]} \arrow[dl] \\
& \sum_{1}^{n} s_i + h_i(H_1) \arrow[d] & \\
& s'' = (s_1'', s_2'' \cdots, s_n'') &
\end{tikzcd}$$

1.16. Optimal Feature Vector Set Generation

In the first part we have obtained a good set of feature vectors by Deep learning algorithm. In this phase we will fine tune the obtained feature vector set by adjusting the weight of the units of the RBM. To fine tune the feature vector set optimally we use back propagation algorithm. Back propagation algorithm is well known method to adjust the deep architecture to find good optimum feature vector set for the precise contextual summary of text. The deep learning algorithm in this phase uses cross-entropy error to fine tune the obtained feature vector set. The cross-entropy error for adjustment is calculated for every feature of the sentence .For example term weight feature of the sentence will be reconstruct by using following formula:

$$[-\sum_{v} f_v \log f_v^{\wedge} \wedge -\sum_{v} (1-f_v)\log(1-f_v^{\wedge})]$$

Where:
f_v = The t_f value of v^{th} word
$f_v^{\wedge}$ = The t_f value of reconstruction

In this way all three features will be optimized.

1.17. Summary Generation

In summary generation phase, the obtained optimal feature vector set is used to generate the extractive summary of the document. For summary generation first task is obtaining the sentence score for each sentence of document. Sentence score is obtained by finding the intersection of user query with the sentence. After this step ranking of the sentence is performed and the final set of sentences for text summary generation defining the summary is obtained.

1.18. Sentence Score

Sentence score ratio of common words found in query of user and particular sentence to the total number of words in the text document. It is given by:

$$S_c = \frac{s \cap Q}{wc}$$

Where:
Sc = Sentence score of a sentence
S = Sentence
Q = User query
Wc = Total word count of a text

1.19. Ranking of Sentence

This is the final step to obtain the summary of text. Here ranking of the sentence is performed on the basis of the sentence score obtained in previous step. The sentences are arranged in descending order on the basis of the obtained sentence score. Out of these sentences top-N sentences are selected on the basis of compression rate given by the user. To find out number of top sentences to select from the matrix we use following formula based on the compression rate.

It is given by:

$$N = \frac{C \times N_S}{100}$$

Where:
N_s = Number of sentences in document
C = Compression rate

1.20. Result and Analysis

The proposed approach deals with text summarization based on a deep learning method. The method that we proposed incorporates the RBM algorithm for getting better efficiency. The performance of the proposed approach is evaluated in the following section 1.21 onwards under different evaluation criteria. All algorithms are implemented in JAVA language and executed on a core i5 processor, 2.1MHZ, 4 GB RAM computer.

1.21. Dataset Description

The experimental evaluation of the proposed text summarization algorithm is executed on different documents. The documents are collected from specific area like data mining, software engineering. Multiple documents from each of the different domains are collected and processed, since the proposed approach is based on multiple documents. The data mining keyword is given in the Google search and the top ten result is selected. The top ten results are stored as ten documents and given to the feature extraction phase to extract the feature vectors. Similarly, the document set for software engineering and networking are created and features are extracted.

1.22. Evaluation Metrics

The evaluation of the proposed text summarization method is based three basic evaluation criteria. The different criteria are listed below.

1.23. Recall

Recall is the ratio of number of retrieved sentence to the number of relevant sentence. The recall is used to measure the reliability of the proposed text summarization method:

$$\text{Recall} = \frac{S_{Ret} - S_{Rel}}{S_{Ret}}$$

where, S_{ret} and S_{rel} are the number of retrieved and relevant sentences respectively.

1.24. Precision

The ratio of retrieved sentences to relevant sentences based on the relevant sentences is given as the precision measure:

$$\text{Precision} = \frac{S_{Ret} - S_{Rel}}{S_{Rel}}$$

1.25. F-Measure

The precision values and the recall values are considered for finding the F-measure value for the total dataset. Thus the F-measure can be expressed as:

$$\text{F} - \text{measure} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}$$

1.26. Feature Vector Extraction

The feature extraction result of the proposed multi-document summarization is explained in section 1.26. Here we have taken ten documents of similar topics as input. The generated summary is then evacuated using the summary available in the dataset by measuring precision, recall and the F-measure. The measurements are then calculated by using different percentage in summary.

The **Table 1** represents the feature vectors extracted from the given set of documents. The represented values are listed based on the highest values possessed from the whole provided data. The values of four features are plotted in the above table, respective of the specific documents.

1.27. Performance Evaluation

The performance evaluation of the proposed approach is discussed in section 1.27. The evaluation process is carried out in three different document sets. The response of the three document set regarding the proposed approach is plotted in the following section 1.28. The recall, precision and f-measure for all the three dataset are calculated by varying different threshold values. The different threshold values are used to verify the responses of the proposed text summarization algorithm under different condition. The threshold is selected from the RBM algorithm. Three filtering threshold for each of the document set are used.

In the **Fig. 3** the response of the document set one is the plotted. The document set consists of documents regarding networking domain. The number of documents included in the document set is ten documents. The summary is generated with the help of the proposed text summarization algorithm. The maximum recall value obtained for the networking domain is 0.429 for filtering threshold 1. Similarly the maximum precision value obtained is 0.6 for threshold. The f-measure value is calculated according to the recall and precision value. The maximum value obtained for the f-measure is 0.490.

The above **Fig. 4** shows the responses of software engineering related data documents. The responses are different as compared to the first set of documents. The maximum recall and precision value for the current dataset is giving as 0.342 and 0.83 respectively. The f-measure value can be listed as 0.469.

The response of the document set, which is related to the networking domain, is plotted in the above **Fig. 5**. Response of the networking domain is also quite different from all other domains. From this analysis, it is clear that the proposed text summarization algorithm is sensitive to the data, which are inputting to the algorithm.

1.28. Comparative Analysis

We plot the comparative analysis of the performance of the proposed approach and an existing method. Both the methods are triggered based on the deep learning algorithm. The algorithm concentrates on the recall values of the proposed approach and the existing approach. The recall values of both the algorithm based on particular datasets have been taken here for the comparative analysis.

The **Fig. 6** shows the comparative analysis of the proposed approach and the existing approach. The recall values plotted in the above graph is taken by varying the threshold values from 0.5 to 2. The analysis from the graph shows that the proposed approach responds better as compared to the existing one.

Table 1. Feature vector extraction

Document no:	Paragraph no:	Line no:	Title value:	Position value:	tf_idf:	Concept:
2	0	1	0.888889	3.0	0.736645722	0.290139693
2	2	1	0.777779	2.8	0.730378687	0.655319108
2	2	2	0.666667	2.6	0.731382288	0.674829335
3	0	1	0.700000	3.0	0.694924858	0.213265682
3	2	3	0.800000	2.4	0.952361002	0.471023052
8	4	3	0.555556	2.4	0.489351427	0.182562528
8	5	1	0.444444	1.0	0.462419924	0.148672137
9	0	1	0.727273	3.0	0.724465870	0.219798458
9	5	2	0.545455	2.6	0.671540289	0.405503813

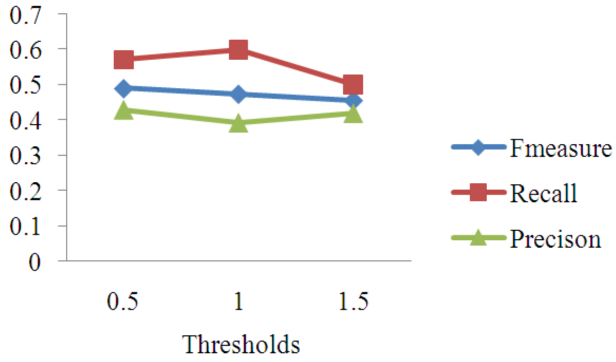

Fig. 3. Performance of networking domain

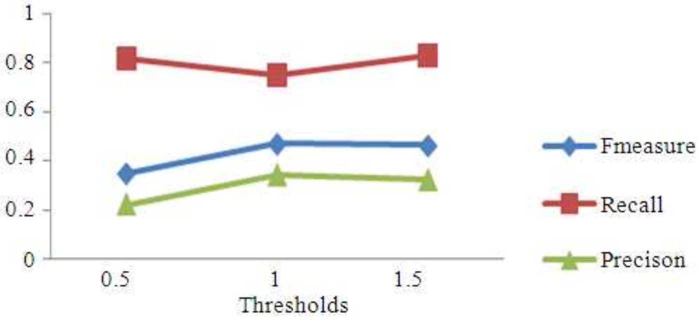

Fig. 4. Performance of software engineering domain

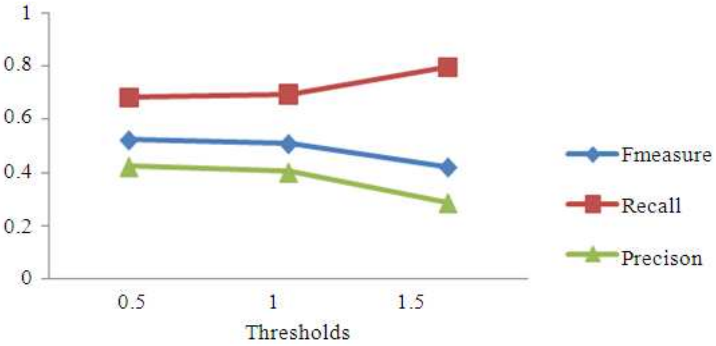

Fig. 5. Performance of networking domain

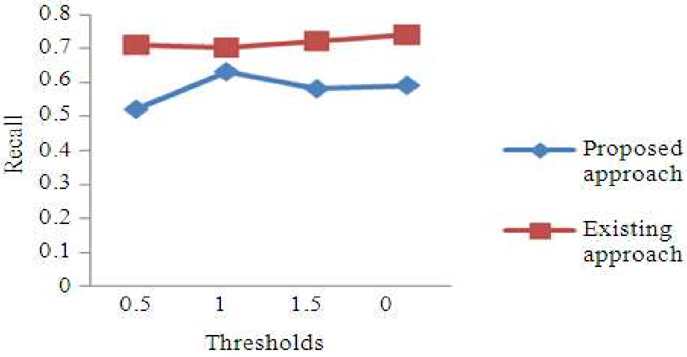

Fig. 6. Comparative analysis

The maximum recall values marked for the existing approach is 0.72, while for the proposed approach it comes around 0.62.

2. CONCLUSION

Several researches were conducted for summery generation from the multiple documents in recent days. We have developed automatic multi-document summarization system which incorporates the RBM. We have used four different features for feature extraction phase. The feature score of the sentences is applied to the RMB in which the RBM rules are optimized with the help of Deep Learning Algorithm. The features are processed through different levels of the RBM algorithm and the text summary is generated accordingly. The generated result is tested as per the evaluation matrices. The evolution matrices considered in the proposed text summarization algorithm are recall, precision and f-measure. The experimentation of the proposed text summarization algorithm is carried out by considering three different document sets. The responses of three documents sets to the proposed text summarization algorithm are satisfactory. The performance judging parameter f-measure has got values, 0.49, 0.469 and 0.520 respectively for the three document sets. The futuristic enhancement to the proposed approach can done by considering different features and by adding more hidden layers to the RBM algorithm.

3. REFERENCES

Darling, W.M. and F. Song, 2011. Probabilistic document modeling for syntax removal in text summarization. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, (CL' 11), ACM Press, Stroudsburg, PA., pp: 642-647.

Goldstein, J., V. Mittal, J. Carbonell and M. Kantrowitzt, 2000. Multi-document summarization by sentence extraction. Proceedings of the NAACL-ANLP Workshop on Automatic Summarization, (WAS' 00), ACM Pres, Stroudsburg, PA, USA., pp: 40-48. DOI: 10.3115/1117575.1117580

Kogilavani, A. and P. Balasubramanie, 2012. Sentence annotation based enhanced semantic summary generation from multiple documents. Am. J. Applied Sci., 9: 1063-1070. DOI: 10.3844/ajassp.2012.1063.1070

Mani, I., 2001a. Automatic Summarization. 1st Edn., John Benjamins Publishing, Amsterdam, ISBN-10: 9027249865, pp: 285.

Mani, I., 2001b. Recent developments in text summarization. Proceedings of the 10th International Conference on Information and Knowledge Management, Nov. 06-11, ACM Press, McLean, VA, USA., pp: 529-531. DOI: 10.1145/502585.502677

Ou, S., C.S.G. Khoo and D.H. Goh, 2008. Design and development of a concept-based multi-document summarization system for research abstracts. J. Inform. Sci., 34: 308-326.

Patil, K. and P. Brazdil, 2007. Text summarization: Using centrality in the pathfinder network. Int. J. Comput. Sci. Inform. Syst., 2: 18-32

Sharef, N.M., A.A. Halin and N. Mustapha, 2013. Modelling knowledge summarization by evolving fuzzy grammar. Am. J. Applied Sci., 10: 606-614. DOI: 10.3844/ajassp.2013.606.614

Zhang, Y., D. Wang and T. Li, 2011. iDVS: An interactive multi-document visual summarization system. Mach. Learn. Know. Disco. Databases, 6913: 569-584. DOI: 10.1007/978-3-642-23808-6_37