ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ASSOCIATION OF DEEP LEARNING ALGORITHM WITH FUZZY LOGIC FOR MULTIDOCUMENT TEXT SUMMARIZATION

[1]G.PADMAPRIYA, [2]Dr.K.DURAISWAMY

[1]Assistant Professor, Department of CSE, K.S.R.College of Engineering
[2] Dean [Academic], Department of CSE, K.S.Rangasamy College of Technology
K.S.R. Kalvi Nagar, Tiruchengode – 637 215,Namakkal[Dt], Tamilnadu, India.

E-mail: 1gpadmapriyame@gmail.com , 2drkduraiswamy@yahoo.co.in

ABSTRACT

Research in text summarization is predominantly targets on measure of the worth of sentences for a summary. The proposed work has associated the Deep learning algorithm with fuzzy logic to improve the efficiency of the generated summary. The proposed work has two phases, they are training phase and testing phases. The training phase utilized to extract the benefits of fuzzy logic and deep learning algorithm for the efficient summary generation. Similar to every training phase, the proposed training phases is also possessed with well known data and attributes. Latter to the training phase, the testing phases is implemented to check the efficiency of the proposed approach. The experimental evaluation of the proposed work provided the predictable results as, the average precision obtained is 0.37, the average recall is 0.86 and the average f-measure is obtained as 0.50%.

Keywords: *Multi-document Summary, Deep Learning, Fuzzy Logic, Natural Language Processing, Neural Network.*

1. INTRODUCTION

With the rapid growth in the quantity and complexity of documents sources on the internet, it has become increasingly important to provide improved mechanism to user to find exact information from available documents. Text summarization has become an important and timely tool for helping and interpreting the large volumes of text available in documents.

Automatic document summarization is the summary of the source version of the original text while keeping its main content and helps the user to quickly understand large volumes of information. Text summarization can handle the problem of selecting the most important portions of text as well as the problem of generating coherent summaries.

Automatic text summarization is significantly different from that of human based text summarization since humans can capture and relate deep meaning and themes of text documents while automation of such a skill is very difficult to implement.

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 in abstractive 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 [8]. In extractive summarization it consider all the sentence in a matrix form, and on the basis of some feature vectors all the necessary or important sentences are extracted.

A feature 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.

In this paper, a method for document summarization is proposed based on deep learning algorithm associated with fuzzy logic. The recent studies have showed that, the deep learning algorithm has more impact on the text summarization process by pointing the most

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

relevant objects from set of objects. In according to the specific behavior of the deep learning algorithm, the proposed approach is plotted.

The proposed text summarization method has two main phases. Feature extraction from multiple documents is considered as the initial phase. The feature extraction includes, feature matrix generation from the features extracted from each sentences. Then, the feature matrix is processed with a fuzzy classifier. The rules generated in the fuzzy classifier are then selected for generating a feature matrix based on fuzzy score. The newly created feature matrix is process through deep learning algorithm and the text summary is generated after layer by layer processing of the learning algorithm.

2. LITERATURE REVIEW

Yan Liu et al [19] have proposed a document summarization framework via deep learning model, which has demonstrated distinguished extraction ability in document summarization. The framework consists of concepts extraction, summary generation and reconstruction validation. A query-oriented extraction technique has been concentrated information distributed in multiple documents to hidden units layer by layer. Then, the whole deep architecture was fine-turned by minimizing the information loss in reconstruction validation part. According to the concepts extracted from deep architecture, dynamic programming was used to seek most informative set of sentences as the summary. Experiments on three benchmark dataset demonstrate the effectiveness of the framework and algorithms.

Jason Weston et al [6] have proposed a supervised learning for deep architectures, if one jointly learns an embedding task using unlabelled data was improved. Researchers used shallow architectures already showed two ways of embedding to improve generalization. First is embedding unlabelled data as a separate pre-processing step (i.e., first layer training) and the second is used for embedding as a regularized (i.e., at the out-put layer). More importantly, they have generalized these approaches to the case where, have train a semi-supervised embedding jointly with a supervised deep multi-layer architecture on any (or all) layers of the network, and showed have been could bring real benefits in complex tasks.

F. kyoomarsi et al [3] have presented an approach for creating text summaries. Used fuzzy logic and word-net, they have been extracted the most relevant sentences from an original document. The approach utilizes fuzzy measures and inference on the extracted textual information from the document to found the most significant sentences. Experimental results reveal that come within reach of extracted the most relevant sentences when compared to other commercially available text summarizers.

Binwahlan et al [14] has incorporated fuzzy logic with swarm intelligence; so that risks, uncertainty, ambiguity and imprecise values of choosing the features weights (scores) could be flexibly tolerated. The weights obtained from the swarm experiments were used to adjust the text features scores and then the features scores and then the features scores were used as inputs for the fuzzy inference system to produce the final sentence score. The sentences were ranked in descending order based on their scores and then the top n sentences were selected as final summary.

Kiani et al[2] proposed a novel approach that extracts sentences based on an evolutionary fuzzy inference engine. The evolutionary algorithm uses GA and GP in concert. The genetic algorithm is used to optimize the membership functions and genetic programming is used to optimize the rule sets. The problem of competing conventions in fuzzy system optimization is thereby reduced by decoupling the two major categories of optimization in fuzzy systems. Fitness function is chosen to consider both local properties and global summary properties by considering various features of a given sentence such as its relative number of used thematic words as well its location in the whole document.

Allan Borra et al. [1] have aimed to develop a system that would be able to summarize a given document while still maintaining the reliability and saliency in the text. To achieve this, two main existing methods such as keyword extraction and discourse analysis based on Rhetorical Structure Theory (RST) have been included in ATS by the system architecture.

3. PROPOSED WORK

For summarizing the text there is a need of structuring the text into certain model which can be

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

given to Restricted Boltzmann Machine (RBM) as input.

3.1 Restricted Boltzmann Machine

RBM is a stochastic neural network (that is a network of neurons where each neuron has some random behavior when activated). It consists of one layer of visible units (neurons) and one layer of hidden units [5]. Units in each layer have no connections between them and are connected to all other units in other layer as shown below in Figure:1.

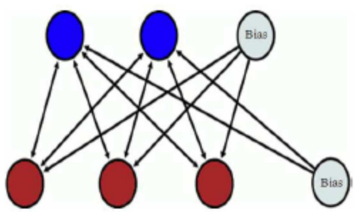

Figure 1 : Restricted Boltzmann Machine

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.

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 which is known as unsupervised learning.

During text summarization the text document is preprocessed using various prevalent preprocessing techniques and then it is converted into feature matrix defined over a vocabulary of words. This feature matrix each row will work as a input to our RBM.

Based on the structured matrix, the proposed text summarization algorithm uses the fuzzy classifier to assign class labels for the sentences, in order to compute the relevance of each sentence based on the rule selector. The rules are then divided into corresponding sentences and the sentences are then used to form the new feature matrix.

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.

The block diagram of the proposed work is given below in Figure: 2

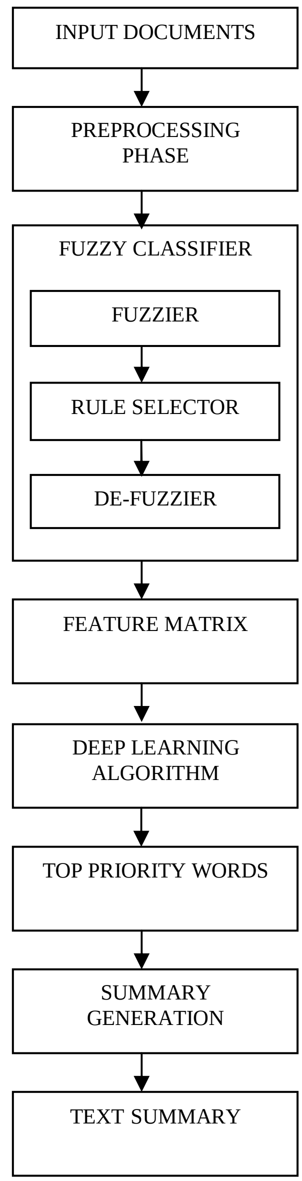

Figure 2 : Overall Block Diagram of Text Summarization

3.2 Preprocessing

Initially, the input to the proposed approach is a set of document from DUC 2002 Dataset that has to be summarized. The document utilized for text summarization is organized by a set of preprocessing steps like, sentence Segmentation, Stop words removal and Stemming.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3.2.1 Segmentation

It is performed by identifying the delimiter commonly denoted by "." called as full stop. This step is used to separate the sentences in the document. It is mainly useful for the user to understand each individual sentence which is there in the document.

3.2.2 Stop Words Removal

Stop words are removed mainly to reduce the insignificant and noisy words. These are predefined words such as a, an, in, by, etc., are called stop words which are filtered out before the preprocessing phase from the documents.

3.2.3 Stemming

Stemming is process of bringing the word to its base or root form for example using words singular form instead of using the plural. It basically removes the prefix and suffix of the concerned word to get the base form. There are more number of algorithms, which are called as stemmers used to perform the stemming process.

3.3 Association of Deep Learning Algorithm with Fuzzy Logic

The proposed algorithm is an association of the Deep learning Algorithm with Fuzzy Logic and it is characterized by two phases. The phases are namely, the training phase and the testing phases. The training phase is used gain the advantages from fuzzy logic and deep learning algorithm to make the text summarization process an effective one. Similar to every training phase, the proposed training phases is also possessed with known data and attributes. Latter to the training phase, the testing phases is implemented to test the efficiency of the proposed approach.

3.3.1 Training Phase

The proposed text summarization process through deep learning algorithm is characterized by the training process of the approach. The main attributes of the text summarization is the features extracted from the multiple documents, which are considered for the summarization process. On behalf of the training phase, the proposed approach defines five features sets. The feature sets are listed as follows,

3.3.1.1 Title Similarity Feature

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 and it is calculated by the formula given below.

$$Title\ Feature\ (f1) = \frac{S \cap t}{t}$$

where,f1 is the features extracted according to the tittle similarity of the documents. S is the set of words extracted by analysing the sentences present in each documents and t is the words extracted from analysing the titles in each documents.

3.3.1.2 Positional Feature

To calculate the positional score of sentence, the proposed approach considers the following conditions. If the sentence given is in the starting of the sentence or the last in the sentence of the paragraph then the feature value f_2 is assigned as 1. Else if the sentence is in the middle of the paragraph then the feature value of f_2 is assigned as

3.3.1.3 Term Weight Feature

The Term Frequency of a word will be given by TF (f, d) where f is the frequency of the given word and d is text present the document. The Total Term Weight is calculated by Term Frequency and IDF for a document .Here IDF denotes the inverse document frequency which just implies that the term is common or rare across all documents. It is obtained by dividing the total number of documents by the number of documents holds the term, and then computing the log of that quotient. The IDF value retrieved by,

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

WHERE, D is the total number of documents $\epsilon D: t\epsilon d$, it is the number of documents in term t appears. The total term weight is given by TFxIDF which can be calculated by

$$f_3 \Rightarrow TF \times IDF(t, d, D) = TF(t, d) \times IDF(t, D)$$

3.3.1.4 Concept Feature

The concept feature from the text document is retrieved 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 have to find out the co-occurrence of words in same window and it can be calculated by following formula,

$$f_4 \Rightarrow MI(w_i, w_j) = \log 2 \frac{P(w_i, w_j)}{P(w_i) \times P(w_j)}$$

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,
$|sw_t| -$ is the number of windows containing the keyword w_i.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$|sw| -$ total number of windows constructed from a text document.

3.3.1.5 POS Tagger Feature

Part of speech tagging is the process of categorizing the words of text on the basis of part of speech category such as noun, verbs, adverb, adjectives, they belong to. Algorithms such as hidden Markova models, using dynamic programming are used to perform this task. The POS Tags on each document is feature five (f_5) of the given documents.

3.3.2 Feature Matrix

Here sentence matrix $S = (s_1, s_2, \dots s_n)$ where $s_i = (f_1, f_2, \dots f_5)$, $i \leq n$ is the feature vector. The five features are the main attributes of the proposed text summarization algorithm. The whole documents under consideration are subjected for the feature extraction and a set of features are extracted accordingly. Now based on the collected features a feature matrix is formed by mapping the features values. The feature matrix is constructed according to the sentences extracted from the multiple documents. In addition to the five features, an additional attribute also associated with the feature matrix. The addition feature associated with the feature matrix is the class labels for each sentence. The Figure 3 represents the feature matrix of the set of documents under consideration.

$$S = \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ \cdots \\ s_n \end{pmatrix} = \begin{pmatrix} f_1 & f_2 & f_3 & f_4 & f_5 & C \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \end{pmatrix}$$

Here, the attribute defined by S represents the sentences and Class represents the class values of each sentences. Usually, the class labels are assigned manually by experts from the domains, but in the proposed approach; we define a fuzzy classifier for assigning the class labels for the sentences. The fuzzy classifier assigns the class labels to the sentences according to the fuzzy rules by processing the sentences.

3.3.3 Fuzzy Logic System

The proposed text summarization algorithm uses the fuzzy logic system has to assign class labels for the sentences, in order to compute the importance of each sentence. The fuzzy logic system accepts the pre summarized set of documents as input. It has 3 main components namely, Fuzzifier, Rule Selector and the Defuzzifier.

The fuzzifier is a process of translating the inputs into feature values. The proposed approach uses a set five features, the fuzzier defines a value in the range 1 to 5 for each features. Based on these fuzzy values, rules are generated for each sentence by the weight given to the features. A rule can be defined for the proposed approach as, a set of features value is considered for judging the importance of sentences. The rules are composed based on the importance of the each sentence. If a feature has value VERY LOW, then it assigns least importance to the sentence, if values is LOW, it assigns low importance to the sentence, if the value is MEDIUM, then it assigns medium importance to the sentence, if the value is HIGH, then it assigns the high importance to the sentence and if the value is VERY HIGH, then it assigns the very high relevance to the sentence. Thus if a sentence is assigned by rule with all five features value are 1, and then the sentence is considered as least important for the summary and vice versa. Thus a set of rules are constructed by comparing the sentences from the set of documents and the sentences from the text summary.

The rule selector is used to select the important rules from the set of rules created by fuzzifier. Here the important rules which are needed for the text summarization are stored in a set.

De-fuzzier is performed finally for data preparation of the deep learning algorithm. Here the defuzzifier selects the needed rules from the rule selector and assign the fuzzy score for each sentences accordingly. Therefore the defuzzifier finally modifies the feature matrix based on the feature values assigned to a particular rule and derives the fuzzy score by evaluating the features values. The rules are then divided into corresponding sentences and the sentences are then used to form the new feature matrix which is the input to the deep learning algorithm.

3.3.4 Deep Learning Algorithm

The sentence matrix $S = (s_1, s_2, \dots s_n)$ which is the feature vector set having element as s_i which is set contains the all the five 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:

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

Journal of Theoretical and Applied Information Technology
10th April 2014. Vol. 62 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

where, $s_i = (f_1, f_2, \dots f_5)$, $i \leq n$ 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 $H_0 H_1$:

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

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 $(s_1, s_2, \dots s_n)$ 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 step 1 is our set of sentence matrix, $S = (s_1, s_2, \dots s_n)$, 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 $s' = (s_1', s_2', \dots s_n')$. 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 $s'' = (s_1'', s_2'', \dots s_n'')$

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.

3.4 Optimal Feature Matrix

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:

$$[-\textstyle\sum_v f_v \log f_v^\wedge - \sum_v (1-f_v)\log(1-f_v^\wedge)]$$

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.

3.5 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.

3.6 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:

$$S_c = \frac{S \cap Q}{W_c}$$

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

3.7 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

4. EVALUATION METRICS

Journal of Theoretical and Applied Information Technology
10th April 2014. Vol. 62 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The following metrics are used to find the efficiency of the proposed system.

4.1 Recall: Recall is the ratio of number of retrieved sentence to the number of relevant sentence. It is used to find the reliability system.

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

Where, S_{Ret} and S_{Rel} are the number of retrieved and relevant sentences respectively.

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

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

4.3 F-measure: The precision values and the recall values are considered for finding the F-measure value for the total data set.

$$F - Measure = \frac{Recall \times Precision}{Recall\ + Precision}$$

5. RESULT ANALYSIS

We have taken few sample documents of similar topics as input. The Optimal Feature Matrix generation for a sample document is shown in Table 1:

Table 1: Optimal Feature Matrix.

Document ID: AP880911-0016						
Sentence ID	f_1	f_2	f_3	f_4	f_5	Class
1	1.0	2.8	1.0	1.0	0.2	1
2	1.0	2.0	0.3	0.5	0.6	1
3	0.0	2.5	0.2	0.5	0.6	0
4	0.0	2.0	0.6	0.5	0.2	0
5	3.0	1.8	0.1	1.0	0.4	0
6	1.0	2.7	0.4	0.5	0.1	1

The generated summary is then evaluated based on the Evaluation metrics such as Recall, Precision and F-Measure. The maximum Recall, Precision and F-Measure values for the current dataset is giving as 0.37, 0.86 and 0.50 respectively.

The Figure: 3 represent the comparison graph between our proposed system and our existing system [5].

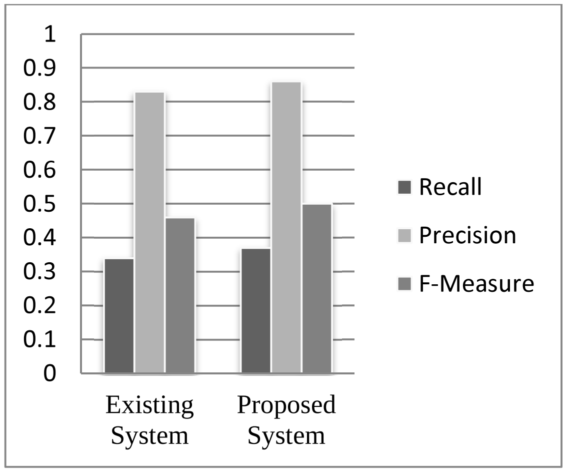

Figure 3 : Comparison Between Existing And Proposed System

6. CONCLUSION

In this proposed work we have extracted five features for feature matrix from the set of sample dataset from DUC2002. The feature matrix is applied to our proposed work which associates the fuzzy logic with deep learning algorithm.

The feature matrix is applied through the different levels of the RBM and finally the efficient text summary is generated. The result analysis shows that the proposed work produce the better performance than the existing work based on the evaluation metrics. The maximum Recall, Precision and F-Measure values for the current dataset of the proposed work is obtained as 0.37, 0.86 and 0.50 respectively for the proposed system.

REFERENCES

[1] Allan Borra, Almira Mae Diola, Joan Tiffany T. Ong Lopez, Phoebus Ferdiel Torralba, Sherwin So, "Using Rhetorical Structure Theory in Automatic Text Summarization for Marcu-Authored Documents", *In titaniaadduedupH*, 2010.

[2] B.Arman kiani, M.R. Akbarzadeh, "Autmatic Text Summarization using: Hybrid Fuzzy GA-GP" *IEEE International Conference on Fuzzy Systems*,2006.

[3] C.Gordon and R.Debus, "Developing deep learning approaches and personal teaching efficacy within a preservice teacher education

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

context," *Web Intelligence and Intelligent Agent Technology, British Journal of Educational Psychology*,Vol. 72, No. 4, 2002, PP. 483-511.

[4] 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. 15-32.

[5] G.Padmapriya, Dr.K.Duraiswamy, " An Approach For Text Summarization Using Deep Learning Algorithm", *Journal of Computer Science*, vol.6, no.11, 2013, PP.1-9.

[6] Inderjeet Mani, "Automatic summarization", *Kluwer Academic Publishers*, 2004.

[7] Jason Weston, Frederic Ratle and Ronan Collobert," Deep Learning via Semi - Supervised Embedding, "*International conference on Machine learning*, 2008, PP. 1168-1175.

[8] Jen-Yuan Yeh , K. Hao-Ren and Wei-Pang Yang , "iSpreadRank: Ranking sentences for extraction-based summarization using feature weight propagation in the sentence similarity network", *Expert Systems with Applications*, vol. 35, no. 3, October 2008, PP. 1451-1462.

[9] Joel Larocca Neto, A Alex, Freitas and A. Celso A. A. Kaestner, "Automatic Text Summarization using a Machine Learning Approach," *Springer Link*, 2012.

[10] Kaustubh Patil and Pavel Brazdil, "SUMGRAPH: Text Summarization Using Centrality InThe Pathfinder Network", *International Journal on Computer Science and Information Systems*, vol.2, no.1, 2007, PP. 18-32.

[11] Khushboo Thakkar and Urmila Shrawankar, "Test Model for Text Categorization and Text Summarization," Web Intelligence and Intelligent Agent Technology, *International Journal on Computer Science and Engineering (IJCSE)*,Vol. 3, No. 4, Apr 2011.

[12] LaddaSuanmali, NaomieSalim and Mohammed Salem Binwahlan, "Automatic Text Summarization Using Feature Based Fuzzy Extraction ",*Jurnal Teknologi Maklumat*, vol. 20, no. 2, Dec 2008, PP. 105-115.

[13] Leonhard Hennig, Winfried Umbrath and Robert Wetzker, "An Ontology-based Approach to Text Summarization," *Web Intelligence and Intelligent Agent Technology*, Vol.3, 2008, PP. 291- 294.

[14] Mohammed Salem Binwahlan, Naomie Salim and Ladda Suanmali, "Fuzzy Swarm Based Text Summarization ", *Journal of Computer Science*,2009, PP. 338-346.

[15] Rajesh Shardanand Prasad and Uday Kulkarni, "Implementation and Evaluation of Evolutionary Connectionist Approaches to Automated Text Summarization", *Journal of Computer Science*, vol. 6, no. 11, 2010, PP. 1366-1376.

[16] Rucha S. Dixit and Prof.Dr.S.S. Apte, "Improvement of Text Summarization using Fuzzy Logic Based Method", *IOSR Journal of Computer Engineering*, Sep-Oct. 2012, Vol. 5 No. 5, PP. 05-10

[17] V. Yatsko, S. Shilov and T. Vishniakov, "A Semi-automatic Text Summarization System", *In proceedings of the 10th International onference on Speech and Computer, Patras*, 2005, PP. 283-288.

[18] W.M. Darling, and F. Song, "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*.2011, PP. 642-647.

[19] Yan Liu, Sheng-hua Zhong, Wen-jie Lii , "Query-oriented Unsupervised Multi - document Summarization via Deep Learning," *Elsevier Science*, 2008, PP. 3306 – 3309.