Advances in Information Mining
ISSN: 0975-3265 & E-ISSN: 0975-9093, Volume 4, Issue 1, 2012, pp.-44-51.
Available online at http://www.bioinfo.in/contents.php?id=32

PERFORMANCE ANALYSIS OF SINGLE DOCUMENT SUMMARIZATION SYSTEMS

SAKHARE D.Y.[1] AND RAJKUMAR[2]

[1]Maharashtra academy of engineering Alandi (D), Pune, Maharashtra, India.
[2]DRDO Scientist 'D', DIAT, Khadakwasla, Pune, Maharashtra, India.
*Corresponding Author: Email- diptiysakhare@gmail.com

Received: March 09, 2012; Accepted: April 03, 2012

Abstract- In recent times, the necessity of generating single document summary has gained popularity among the researchers due to its extensive applicability. Most of the automatic text summarization systems utilize extraction-based techniques for selecting the most significant portions of text to generate coherent summaries. In this paper we will analyze the performance of fuzzified neural network approach with the graph theory approach. In the proposed system, we have developed an efficient automatic text summarization system based neural network and fuzzy logic. In the training phase at first, the feature vector is computed for a set of sentences using the feature extraction technique. After that, the feature vector and their corresponding fuzzy score are used to train the neural network optimally. Later in the testing phase, the input document is subjected to preprocessing and feature extraction techniques. In order to obtain the sentence score for every sentence in the input document, the feature vector is fed to the trained neural network that returns the sentence score for every sentence. Finally, by making use of sentence score, the most important sentences are extracted from the input document. The experimentation is performed with the DUC 2002 dataset and the generated summary is evaluated with the measures such as Precision, recall and f-measure. The comparative results of our proposed approach with the graph theory approach produces better results by means of different compression rates.
Key words- Text summarization, Feature extraction, Multi-layer Perceptron Neural Network (MLPNN), Fuzzy logic, fuzzy score, DUC 2002 dataset

Citation: Sakhare D.Y. and Rajkumar (2012) Performance Analysis of Single Document Summarization Systems. Advances in Information Mining, ISSN: 0975-3265 & E-ISSN: 0975-9093, Volume 4, Issue 1, pp.- 44-51.

Introduction

A significant and opportune tool that assists and interprets huge quantities of text presented in documents is text summarization [1]. The process of making a summary of one or more texts is called text summarization and it is used for various purposes [14]. The objective of text summarization is to make a brief version of the original text with the most significant information at the same time retaining its main content and to enable the user to quickly comprehend huge quantities of information [8]. Text summarization tackles the problem of selecting the most significant sections of text as well as the problem of producing organized summaries [20]. Summaries may be created for a single document or for multiple documents. Single document summarization systems process documents one at a time, on the other hand multi-document summarization systems simultaneously process more than one document.Multi-document summarization is normally used to summarize thematically related documents [2]. Nowadays, enormous amount of digitally stored information is available even for many insignificant languages. So in order to prevent sinking in it, filtering and extraction of information are necessary [14].

Normally document summaries are of two types, namely, generic and query-dependent (user-focused). User-focused summaries contain information most relevant to the initial search query, whereas generic summaries contain information about the overall perception of the documents' content. Extensive coverage of document topics and low redundancy must be maintained by generic summaries [4], [5], [16]. Textual coherence, a significant feature of the summary quality, is normally achieved in highly compressed summaries by compromising topic coverage. This is one of the reasons for which summarization researchers overlook topic coverage. But, in certain cases, retaining textual coherence is given more precedence [15]. Document summarization, was generally done by humans. But recent information overload problem has led to the development of automated text summarization systems which solves the problem successfully by drastically compressing the information content [3].

Automatic summarization techniques utilizing a computer summarize a longer text to a shorter form without redundancy [14]. Abstraction and extraction are the two types of summary [7]. Abstraction summary methods generate abstracts by examining and interpreting the text utilizing linguistic methods. Extractive summarization methods select the best-scoring sentences from the original document based on a set of extraction criteria and present them in the summary [12] [18]. Extraction methods are extensively used nowadays for generating the summary by most of the automated text summarization systems [8], [9], [6], [1], [10]. Extractive summarization algorithms, normally based on sentence extraction techniques, attempt to identify the set of sentences that are vital for the overall understanding of a given document [10], [11], [17].

Nowadays automatic text summarization is utilized in a variety of applications, including search engine hit summarization (summarizing the information in a hit list fetched by certain search engine); physicians' aids (to summarize and compare the prescribed treatments for a patient); creating the brief of a book and so on [13]. Best performance of automatic text summarization is achieved if the document is well-structured, for example news, reports, articles and scientific papers [19]. Normally, automatic document summarization accepts one or more source documents as input and provides an elegant summary as output to the user by extracting the gist of the source(s). The process consists of three phases, namely, analysis, transformation and synthesis. In the analysis phase, a small number of significant features are chosen by analyzing the input document. In the transformation phase a summary corresponding to the user's need is generated by transforming the output of the analysis phase. Compression rate, defined as the ratio of summary length to original length, is a significant factor that influences the overall quality of the summary. Increased compression rate results in more voluminous summary consisting of comparatively more unimportant information [17].

Here, we have devised an efficient and effective system for automated text summarization that combines the neural network model and fuzzy techniques. Initially, the set of sentences is prepared by a set of preprocessing steps namely, sentence segmentation, tokenization, stop words removal and word stemming. The preprocessed sentences is subjected to feature extraction process so that, the feature vector is computed for each sentence. Then, by making use of genetic programming (GP), large number of feature vectors (chromosomes) are generated iteratively utilizing cross over and mutation operators. Then, the chromosomes with their fuzzy score are fed to the neural network for training. In the testing phase, the input text document is preprocessed and the feature score of every sentence in the document is computed. The computed feature score is applied to the trained network which returns the final score for each sentence present in the input text document. Based on the computed score value, the coherent and correctly-developed summary is generated for the given input text document.

The rest of the paper is organized as follows: The review of recent researches related to the automated text summarization is given in Section 2. The proposed system architecture for the automatic text summarization is presented in section 3. The experimental results and analysis is presented in Section 4 and the conclusion is given in Section 5.

Related Work

A lot of researches are available in the literature for automatic text summarization of single documents. Recently, a handful of researches have been presented for automatic text summarization system based on artificial Intelligence and evolutionary techniques. Some of the works presented for automatic text summarization are given below:

M. S. BinWahlan et al. [28] have introduced automatic text summarization method based on an integrated hybrid model. Their model attempts to exploit the strengths of different techniques. They have utilized diversity-based method to select the most diverse sentences by filtering similar sentences. Swarm-based method has been used to differentiate between more important and less important features. Finally, they have used fuzzy logic to flexibly tolerate the risks, uncertainties, ambiguities and imprecise values of the text feature weights. Reducing redundancy problems was the focus of diversity-based method whereas the focus of the other two techniques was on the sentence scoring mechanism. Experimental results have proved that their proposed method utlizing the combination of diversity measures, swarm techniques and fuzzy logic was capable of generating good summary by selecting the most important parts of the document.

Carlos Mendez Cruz and Alfonso Medina Urea [22] have detailed an unsupervised approach for automatic summarization. Their concept is to rate each sentence of the document based on the information content of the graphical words it contains. Moreover, they have included a sentence position coefficient as a basic measure of document structure. M.S. Binwahlan et al. [23] have utilized particle swarm optimization to analyze the effect of the feature structure on features selection. Utilizing DUC 2002 data particle swarm optimization has been trained to learn feature weights. In addition to simple or individual features, features formed as a combination of more than one feature having different structures were also used. Accordingly, features having high importance were differentiated from those having low importance by determining the effectiveness of each type of feature. They have assumed that in selection the priority of combined features is more than that of simple features. Experimental results have confirmed that combined features were more effective than simple features.

L Antiqueira et al. [24] have proposed a method for extractive summarization which utilizes the concepts and metrics of complex networks for selecting the sentences. A piece of text was represented as a graph or network in which, sentences were represented as nodes and sentences that share common meaningful nouns were represented as edges. It has been confirmed that the complex networks representation of texts enabled automatic summarization by capturing important text features utilizing network metrics as expected. S. BinWahlan et al. [25] have integrated fuzzy logic with swarm intelligence to adjustably put up with risks, uncertainty, ambiguity and imprecise values of choosing the features weights (scores). The text features scores were adjusted utilizing the weights obtained from the swarm experiment. Final sentence score was generated by giving the adjusted features scores as input to the fuzzy inference system. The sentences were arranged in descending order of their score and the final summary was produced by selecting the top n sentences. The experiments confirmed that most important sentences were included in the final summary because of the vital role played by the fuzzy logic incorporated swarm intelligence in the selection process. Moreover the results confirmed that the performance of the proposed method was good and superior to that of swarm model and benchmark methods.

Ladda Suanmali et al. [1] have concentrated on extraction approach based text summarization. Sentence selection was the objective of extraction approach based text summarization. One of the methods utilized sentence weighting by which some numerical measure was assigned to each sentence to select the best sentences for the summary. Identification of important features was the first step of summarization by extraction. In order to improve the quality of the summary created by the general statistic method, they have proposed fuzzy logic based text summarization. The results have confirmed that summaries with highest average precision, recall, and F-measure were produced by fuzzy method. Ladda Suanmali et al. [26] have proposed an improved feature scoring technique based on fuzzy logic for producing good summary. The only way for determining the important ideas in the text for creating the text summary is text features scoring mechanism. Good summary could be produced by scoring the text sentences utilizing efficient techniques. Inaccurate and unsure feature scores make it difficult to distinguish between important and unimportant features. They have proposed to address the problem of inaccurate and unsure feature score utilizing fuzzy logic. The results have verified that fuzzy method obtained highest average precision, recall, and F-measure for the summaries.

M. S. BinWahlan et al. [27] have addressed the automatic text summarization problem by introducing an intelligent model. Their model attempts to exploit the strengths of different techniques. They have utilized diversity-based method to select the most diverse sentences by filtering similar sentences. They have proved the hypothesis that a combined intelligent model could produce a good summary by exploiting the advantages of different resources. The importance of such intelligent model in solving the automatic text summarization problem was verified by the results obtained by the proposed model. Kaikhah, K. [21] has presented a summarizing technique utilizing neural networks for summarizing news articles. A neural network was trained to learn the significant features of sentences that are suitable for inclusion in the article summary. Then the significant features are generalized and combined and the neural network was modified accordingly. After modification the neural network acts as a filter and summarizes news articles.

Text Summarization System Based on Neural Network, and Fuzzy Techniques

Text summarization has become a significant and well-timed tool for accommodating and interpreting huge amount of text existing in documents. Text summarization is an automatic process that creates the shortened version of a text. Recently, several researches [21-28] have effectively used automated methods for generating a relevant, short and fluent summary from the input text documents. In this research, we have developed an automated text summarization system utilizing evolutionary connectionism. Figure 1 depicts the proposed system architecture for text summarization based on evolutionary, connectionist, and fuzzy techniques.

The proposed automatic text summarization system consists of the following components:

- Preprocessing
- Feature extraction
- Model building

Sentence selection and assembly

a. Preprocessing

Preprocessing is the first component of the system with three different phases: sentence segmentation, removing stop words and, stemming. After applying preprocessing techniques, individual sentences and their unique ID are obtained from the text document

- Segmentation process is achieved by finding out the delimiter ("." full stop) so that, the sentences in the document are separated. This enables a user to recognize every individual sentence available in the document.
- Stop words [31] are detached from the document during the feature extraction step since they are considered as unimportant and contain noise. Stop words are predefined and are stored in an array and the array is utilized for comparison with the words in the provided document. Once the process of stop word removal is completed, the document is divided into individual words to proceed with the word stemming process.
- Word stemming [30] converts every word into its root form. Word stemming is practically removing the prefix and suffix of the specified word which in turn becomes applicable for comparison with other words.

b. Feature extraction

The text document (D) after preprocessing is subjected to feature extraction by which each sentence in the text document obtains a feature score based on its importance. The text document is represented by set, $D = \{S_1, S_2, \ldots, S_k\}$ where, S_i signifies a sentence contained in the document D . The important text features used in the proposed system are: (1) Word similarity among sentences (2) Word similarity among paragraphs (3) Iterative query score (4) Format based score (5) Numerical data (6) Cue-phrases (7) Term weight (8) Thematic feature (9) Title feature. Every sentence in the text document along with its unique ID has a feature vector consisting of nine elements corresponding to the aforesaid features. Each feature score ranges between 0 and 1.

Word similarity among sentences

A sentence (S) obtains a score based on the number of times the words or terms occurring in other sentences of the document. All the sentences are subdivided into individual words and the subdivided words are compared with the words in the other sentences of the document. The occurrence count of a word (C_w) is the number of other sentences in which the specified word occurred. To attain the sentence occurrence count (C_s), the occurrence count of all the individual words in the sentence S is added together. The ratio of the sentence occurrence count of a specified sentence (S) to the maximum sentence occurrence count in the document is calculated to know the word similarity among sentences (WSS_F1) as well as the score for the feature.

$$C_s = \sum_{i=1}^{n} C_w(i)$$

$$WSS\_F1 = \frac{C_s}{\underset{j \in D}{Max}\,(C_s(j))}$$

where, n number of words in the sentence (S)

Word similarity among paragraphs

The whole paragraph (P) is utilized to extract this feature, instead of individual sentences. Thus a same score will be provided to each sentence that comes within the same paragraph. This feature is equivalent to the word similarity among sentences, and the number of paragraphs in the document which comprises the identical terms or words as in the specified paragraph is known as the paragraph occurrence count (C_p) of the specified paragraph. The ratio of the paragraph occurrence count of the given paragraph to the maximum paragraph occurrence count in the document is thus computed for the score of the word similarity among paragraphs (WSP_F2).

$$WSP\_F2 = \frac{C_p}{\underset{j \in D}{Max}\left(C_p(j)\right)}$$

Iterative query score

The score analogous to this feature is attained by the following three phases: (i) Initial keyword identification (ii) Scoring sentences based on iterative query. Initial keyword identification: We compute the frequency of the words or terms contained in the document and the words are sorted based on its frequency. Then, the top 'n' frequent words are chosen from the sorted list which is known as initial keyword set. Scoring sentences based on iterative query: Query is used to search for a keyword in the given text document and retrieve the sentences which contain the keyword. A tag named count is added to all the sentences in the document and initialized to zero, and then the text document is searched for keywords. The sentences that contains the query terms are identified and the tag relevant to those sentences are incremented. Again the keywords are extracted by finding the frequency among the identified sentences. Thus, the updated key words are then used to identify the sentences for the next iteration. For every iteration, the tag count of each sentence is updated based on the query results. The iteration is terminated when the user specified number of loops is executed or if there is no change in the extracted keyword list. The ratio of the tag count (C_T) to the total number of iterations (T_I) is thus computed for the score of the iterative query score feature, IQ_F3 .

Format based score

Concept-based feature: Initially, the concept is extracted from the input document using the mutual information and windowing process. A windowing process is carried out through the document, in which a virtual window of size ' k ' is moved from left to right until the end of the document. Then, the following formulae are used to find the words that co-occurred together within each window.

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

Where, $P(w_i, w_j) \rightarrow$ The joint probability that both keyword appeared together in a text window

$P(w_i) \rightarrow$ The probability that a keyword w_i appears in a text window

The probability $P(w_i)$ is computed based on $\frac{|sw_t|}{|sw|}$, where sw_t is the number of sliding windows containing the keyword w_i and $|sw|$ is the total number of windows constructed from a text document. Similarly, $P(w_i, w_j)$ is the fraction of the number of windows containing both keywords out of the total number of windows. Then, for every concept extracted, the concept weight is computed based on the term weight procedure and the sentence score is also computed as per the procedure described in term weigh-based feature computation.

Numerical data

The significance stats regarding the core intention of the document is usually reflected by the numerical data within the sentence and this has its own impact on the core idea of the document that quite naturally leads to summary selection. The ratio of the number of numerical data (N_D) that occur in sentence over the sentence length ($\mathrm{S_L}$) is thus used to compute the score for this feature (NU_F5).

Cue-phrases

In general, the phrases such as "in summary", "in conclusion", and superlatives such as "the best", "the most important", "according to the study", "hardly" can be good indicators of significant content of a text document. Here, high score is given to the sentences that contain cue words/phrases. For computing the score for this feature, we first listed a set of cue words in a text file and then, the words in the text document are compared with the words in the predefined list. The sentence score based on the cue phrases is calculated by:

$$CP\_\mathrm{F6} = \frac{\mathrm{CP_S}}{\mathrm{CP_D}}$$

where, $\mathrm{CP_S}$ Number of cue-phrases in the sentence

CP_D Total number of cue-phrases in the document

Term weight

Term weight (TW_F7) is a feature value which is used to identify the salient sentences for summarizing the text documents. The term weight of a sentence is calculated as the ratio of the sentence weight (W_S) to the maximum sentence weight in the given text document. The sentence weight is the summation of the weight factor of all the words in a sentence. The weight factor is the product of word frequency and the inverse of the sentence frequency .

$$TW\_F7 = \frac{W\_S}{\underset{j \in D}{Max}\left(W\_S(j) \right)}$$

$$W\_S = \sum_{i=1}^{n} W_i$$

$$W_i = TF \times ISF$$

$$ISF(t) = log(N_S / N_S(\mathrm{w}))$$

Where, W_S à Sentence weight

W_i à Weight factor of the word in a sentence

n à Number of words in a sentence

TF à The number of occurrences of the term or word in a text document

ISF à Inverse Sentence Frequency

N_S à Total number of sentences in a document

$N_S(\mathrm{w})$ à Total number of sentences that contain the word (w)

Thematic features

Thematic words are nothing but most recurrent words in the specified document. Thematic feature is usually noted as significant due to the reason that the terms which occur recurrently in a document may be associated with the core idea of the document. The number of thematic words signifies the words or terms with maximum possible relativity. As thematic words, the n_f frequent content words at the top are considered. Using the below formula, the score for this feature is calculated.

$$TH\_F8 = \frac{TH_s}{\underset{j \in D}{Max}\left(TH_s(j) \right)}$$

where, TH_s à Number of thematic words in the sentence.

Title features

A sentence is given a high score only if the given sentence contains the title words. The intention of the document is expressed via the word belonging to the title if available in that sentence. The ratio of the number of words in the sentence that occur in title (n_T) to the total number of words in the title (N_T) helps to calculate the score of a sentence for this feature (TT_F9).

c. Generating fuzzy score for candidates using fuzzy logic model

Once the feature extraction is finished, the next step will be the application of fuzzy logic to the for finding the fuzzy score. Fuzzy logic was introduced by Zadeh in the late 1960s [32] and is known as the rediscovery of multivalued logic devised by Lukasiewicz. In fuzzy logic, the truth values of the variables can take any value in the range 0 to 1 (E.g. 0.23), in contradiction of boolean logic, in which variables can be either 1 or 0. The Fuzzy logic system consists of four parts: (1) Fuzzifier (2) Rule base (3) Inference engine and (4) Defuzzifier

Fuzzifier

The feature score of every sentence is given to the fuzzifier, which converts the numerical data into the linguistic values (High, medium, low) using the membership function. The membership function is a curve that describes how each feature score is converted into a membership value (or degree of membership). Here, we use the triangle membership function which is defined as follows,

$$f(x:a,b,c) = \begin{cases} 0 \ , \ if \ x < a \ , \mathrm{x} > \mathrm{c} \\ \dfrac{x-a}{b-a} \ , \ if \ a \le x \le b \\ \dfrac{a-x}{b-a} \ , \ if \ b \le x \le c \end{cases}$$

Where a, b and c are characteristic parameters of a fuzzy set.

Rule base

After fuzzification of the candidate feature vectors, we define the fuzzy rules which are important for any fuzzy logic system. Defining the fuzzy IF-THEN rules serves as the major significant module in any fuzzy system. Antecedent and consequent are the two parts of a rule. Antecedent refers to the probable input feature values and consequent is the inference of the rule that decides whether the sentence is important, average or unimportant on the basis of the input. An example to fuzzy rule is as follows: IF (Word similarity among sentence is H) and (Word similarity among paragraph is H) and (Iterative query score is H) and (Format based score is M) and (Numerical data is M) and (Cue-phrases is H) and (Term weight is L) and (Thematic feature is H) and (Title feature is H) THEN (Sentence is important).

Inference engine

The output of the fuzzifier is fed to the inference engine which in turn compares that particular fuzzy input with the Knowledge base. As a result the importance of a sentence is determined and thus the output of inference engine is one of the linguistic values from the following set {Unimportant, Average, and Important}.

Defuzzifier

The inference engine outputs the linguistic values that in turn are converted by the defuzzifier as crisp values. The crisp value signifies the fuzzy score of the sentences in the document.

d. Neural network model

This section describes the model used in the proposed approach for automatic text summarization. Normally, neural networks are a great deal the most frequently used connectionist model at present. A lot of research using neural networks is made under the more common name "connectionist". Here, we have used the Multi-layer Perceptron Neural Network (MLPNN). A Feedforward Neural Network, is defined as “an artificial neural network in which the directed graph that shows the interconnections between the units does not have any closed path or loop” [38]. A multilayer perceptron is a feedforward artificial neural network model that has at least one layer in-between the input and the output layer. A neural network MLP couples, through functions and weights, certain variables (called inputs) with certain other variables (called outputs) [37]. The neural network used in the proposed system is configured with ten inputs, n hidden and one output layer. The configurations of the network used for our approach is shown in the figure 2.

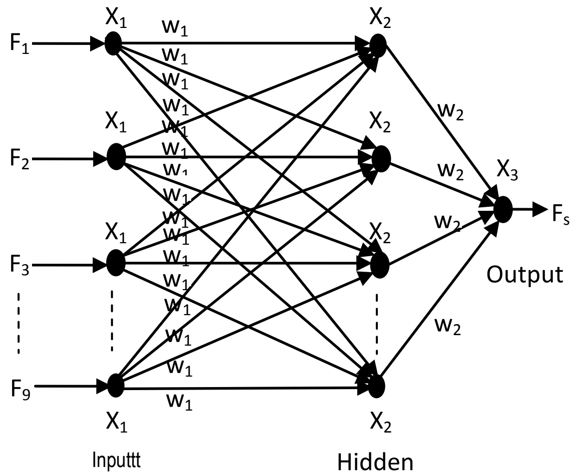

Fig.1- Structure of Multi-layer Perceptron Neural Network

Training phase: The back-propagation algorithm can be utilized successfully to train neural networks; it is extensively accepted for applications to layered feed-forward networks, or multi-layer perceptrons [29]. In order to train the neural network optimally, the input layer is an individual (feature vector) obtained from the EP and the target output is the fuzzy score of the relevant individuals. So, for training the neural network, we make use of evolutionary and artificial intelligence techniques. Testing phase: In testing phase, the input text document is preprocessed and the feature score of every sentence in the document is computed. The computed feature score is applied to the trained network that returns the final score of every sentence presented in the input text document. Based on the computed score value, the coherent and correctly-developed summary is generated for the given input text document.

e. Sentence Selection And Assembly

Two important steps are involved in the selection process of a sentence 1) determining the number of sentence that must be present in the summary based on compression rate and 2) appropriate sentence extraction for the summary. The number of sentences (N) to be placed in the summary is calculated as,

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

Where, N_S Total number of sentences in the document

C Compression rate

Based on the crisp output value from defuzzifier, sentence extraction is attained by arranging the sentence at first in the descending order and thereby the top N sentences are chosen for the summary. A summary has to possess a comprehensible structure and should be presented in a logical manner. On the basis of the order of the reference number or unique ID, the sentences are sequentially ordered to get the final summary.

Experimental Rsults and Analysis

The experimental results and analysis of the proposed automatic text summarization system is presented in this section. The proposed system is implemented in MATLAB (Matlab7.8). We have used DUC 2002 dataset [39] in the proposed system for generating the single document summary. DUC 2002 dataset contains documents on different categories and extractive summary per document.

a. Experimental Results

The experimentation is performed in two different phases namely, training phase and testing phase. Training phase: In the proposed system, as a training data, we have taken 100 sentences from the DUC 2002 dataset (Document No: AP880916-0060, AP900322-0112, AP890607-0067 and LA122190-0149). And then, we apply the preprocessing and feature extraction techniques on the training data so that, we obtain the 100 feature vectors. The sample feature score of the text document (Document No. AP880314-0110) is shown in table 1.

Table 1- Feature score for the text document (Document No. AP880314-0110)

Sentence ID	Feature score								
	F1	F2	F3	F4	F5	F6	F7	F8	F9
S1	1	1	1	0.2500	0	0.3596	1	1	0.2739
S2	0.8571	1	1	0.2500	0.0121	0.2895	0.8	0.5	0
S3	0.5714	1	1	0.2500	0	0.2982	0.4	0	0
S4	0.2857	1	1	0	0	0.2895	0.2	0	0
S5	0.5714	1	0	0.2500	0	0.3158	0.6	0.25	1
S6	0.5714	1	1	0	0	0.307	0.6	0.25	0
S7	0.7143	1	1	0.2500	0	0.3509	0.4	0.75	0
S8	0.5714	1	1	0.2500	0	0.3509	0.6	0.5	0

These feature vectors are fed as an input to the fuzzy logic model that provides the fuzzy score for every vector. The fuzzy score obtained for the text document (Document No. AP880314-0110) is shown in table 2.

Table 2- Fuzzy score for the text document (Document No. AP880314-0110)

Sentence ID	Fuzzy score
S1	0.5095
S2	0.5078
S3	0.5082
S4	0.5178
S5	0.5082
S6	0.5082
S7	0.5086
S8	

The feature vectors chosen from the EP model and their corresponding fuzzy score are used for better training of the neural network. We have used the Multi Layer Perceptron Neural Network which contains nine input layer and one output layer. Testing phase: The input document is taken from the dataset and the pre-przocessing and feature extraction techniques are applied on the input document. The feature score obtained for the input document (Document No. LA080890-0078) is given in table 3.

Table 3- Feature score for the text document (Document No. LA080890-0078)

Sentence ID	Feature score								
	F1	F2	F3	F4	F5	F6	F7	F8	F9
S1	0.8571	1	1	0.4	0	0.307	1	1	0
S2	0.8571	1	1	0	0	0.3509	0.6667	0	0
S3	0.8571	1	1	0	0	0.3684	0.5	0	1
S4	1	1	1	0	0	0.3333	0.6667	0	0
S5	0.8571	1	1	0	0	0.3684	0.3333	0	0
S6	0.8571	1	1	0	0	0.3509	0.6667	0	0
S7	0.1429	1	0	0	0	0.3509	0	0	0.6647
S8	0.8571	1	1	0	0	0.3596	0.5	0	0.5775

The feature score is then directly applied to the trained neural network which returns the sentence score for every sentence in the document. The sentence score obtained from the neural network for the input document is given in table 4. Finally, the salient sentences are extracted by inputting the compression rate.

Table 4- Sentence score for the text document (Document No. LA080890-0078)

Sentence ID	Sentence score
S1	0.6129
S2	0.6108
S3	0.5138
S4	0.5876
S5	0.5862
S6	0.6108
S7	0.5228
S8	0.5597

b. Evaluation Measure

The performance of the proposed approach is evaluated using precision, recall and F-measure [40, 41 and 42]. Precision evaluates the proportion of correctness for the sentences in the summary whereas recall is utilized to evaluate the proportion of relevant sentences included in summary. For precision, the higher the values, the better the system is in omitting irrelevant sentences. Conversely, the higher the recall values the more successful the system would be in fetching the relevant sentences. The weighted harmonic mean of precision and recall is called as F-measure.

$$Precision = \frac{|\{\text{Retrieved sentences}\} \cap \{\text{Relevant sentences}\}|}{|\{\text{Retrieved sentences}\}|}$$

$$Recall = \frac{|\{\text{Retrived sentences}\} \cap \{\text{Relevant sentences}\}|}{|\{\text{Relevant sentences}\}|}$$

Where, Relevant Sentences- Sentences that are identified in the human generated summary

Retrieved Sentences- Sentences that are retrieved by the system

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

c. Comparative Analysis and Performance Evaluation

The performance of the proposed system is evaluated on the summary available in the DUC 2002 dataset using the evaluation measures described in the above section. We have taken four documents from the dataset, D1 (AP880310-0062), D2 (AP880622-0184), D3 (AP880816-0135) and D4 (FT923-5835). Then, we generate the single document summary for these four documents using the proposed system. For experimental analysis, the summary is generated for different compression rate and the generated summary is evaluated on the extractive summary provided in the dataset using the evaluation measures such as, precision, recall and F -measure. The performance of our proposed approach is compared with that of our previous graph theory based approach to automatic text Summarization [44].

We have analyzed our proposed approach with different evaluation measures such as precision, recall and F-measure by different compression rates. The table 5 lists the values for the evaluation measures with compression rate C=40 of our proposed approach and the graph theory approach. The comparative graph of the precision, recall and F-measure is shown in figure 3, 4 and 5 respectively. By analyzing the graphs with C=40, the precision measure of our approach shows better performance for all the four documents D1, D2, D3 and D4. The recall measure and F-measure of the graph theory approach gives better result for the document D2, whereas the documents D1, D3 and D4 of our proposed approach gives improved performance with that of the graph theory approach.

Conclusion

We have developed automatic text summarization system which combines ANN (Artificial Neural Network) and fuzzy logic. Here, we have used nine different features for feature extraction phase. Then the feature vectors are given to the fuzzy logic system so that, the fuzzy score is calculated. The feature vector and their relevant fuzzy score are utilized as a training parameter for training the neural network. In the testing phase, the features extracted from the input text document are given to the trained network that provides score for every sentence in the input document. Finally, we extract the relevant sentences from the input text document in accordance with their sentence score. We have used DUC 2002 dataset to evaluate the summarized results based on the measures such as Precision, recall and f-measure. The experimental results showed that the proposed summarization system effectively summarizes the text documents. In future we will try to develop a new summarization system which will consider the syntactical knowledge along with the existing features

References

[1] Ladda Suanmali, Naomie Salim and Mohammed Salem Binwahlan (2009) *International Journal of Computer Science and Information Security*, 2(1).

[2] Stergos Afantenos, Vangelis Karkaletsis and Panagiotis Stamatopoulos (2005) *Artificial Intelligence in Medicine*, 33, 157-177.

[3] Alkesh Patel, Tanveer Siddiqui and Tiwary U.S. (2007) *Conference RIAO*.

[4] Ramiz M. Aliguliyev (2007) *Journal of Computational Technologies*, 12, 5-15.

[5] Goldstein J., Kantrowitz M., Mittal V., Carbonell J. (1999) 22*nd Annual Intern. ACM SIGIR Conf. on Research and Development in Information Retrieval*, 121-128.

[6] Yulia Ledeneva, Alexander Gelbukh and René Arnulfo Garcia Hernandez (2008) *CICLing*. *LNCS*, 4919, 593-604.

[7] Krysta M. Svore, Lucy Vanderwende and Christopher J.C. Burges (2007) *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 448-457.

[8] Suanmali L., Salim N. and Binwahlan M.S. (2010) *J. Applied Sciences*, 10, 166-173.

[9] Ko Y. and Seo J. (2008) *Pattern Recognition Lett. Arch*., 29, 1366-1371.

[10] Aliruliyev R.M. (2009) *Expert Syst. Appl. Int. J. Arch*., 36, 7764-777.

[11] Rada Mihalcea and Paul Tarau (2005) *IJCNLP*.

[12] Massih R. Aminiy, Anastasios Tombros, Nicolas Usuniery, Mounia Lalmas and Patrick Gallinariy (2005) 14*th ACM international conference on Information and knowledge management*, 297-298.

[13] Kedar Bellare, Anish Das Sarma, Atish Das Sarma, Navneet Loiwal, Vaibhav Mehta, Ganesh Ramakrishnan, Pushpak Bhattacharyya (2004) *International Conference on Language Resources and Evaluation*.

[14] Hassel M. (2007) *Technical report, School of Computer Science and Communication*.

[15] Tristan Miller (2003) *Technical report, University of Toronto*.

[16] Ramiz M Aliguliyev (2006) *IEEE/WIC/ACM international conference on Web Intelligence and Intelligent Agent Technology*, 626 -629.

[17] Jen-Yuan Yeh, Hao-Ren Ke, Wei-Pang Yang, I-Heng Meng, (2005) *Information Processing and Management*, 41(1), 75-95.

[18] Rada Mihalcea (2005) *ACL on Interactive poster and demonstration sessions*, 49-52.

[19] Ladda Suanmali, Mohammed Salem Binwahlan, Naomie Salim (2009) *Ninth International Conference on Hybrid Intelligent Systems*, 1, 142-146.

[20] Brunn M., Chali Y. and Pinchak C.J. (2001) *Workshop on Text summarization in conjunctions with the ACM SIGIR Conference*.

[21] Kaikhah (2004) *2nd International IEEE Conference on Intelligent Systems*, 1, 40-44.

[22] Carlos Mendez Cruz and Alfonso Medina Urrea (2005) *CICLing, LNCS*, 3406, 653-656.

[23] Binwahlan M.S., Salim N. and Suanmali L. (2009) *International Journal of Computer Science and Network Security*, 9(1).

[24] Lucas Antiqueira, Osvaldo N. Oliveira, Luciano da Fontoura Costa and Maria das Graças Volpe Nunes (2009) *Information Sciences an International Journal*, 179(5), 584-599.

[25] Binwahlan M.S., Salim N. and Suanmali L. (2009) *Journal of Computer Science*, 5, 338-346.

[26] Ladda Suanmali, Mohammed Salem Binwahlan, Naomie Salim, (2009) *Ninth International Conference on Hybrid Intelligent Systems*, 1, 142-146.

[27] Binwahlan M.S., Salim N. and Suanmali L. (2009) *Information Technology Journal*, 8, 1249-1255.

[28] Binwahlan M.S., Salim N. and Suanmali L. (2010) *Information Processing & Management*.

Table 5- Precision, Recall and F-measure for compression rate C=40

	|Retrieved sentences|		|Relevant sentences|		|Retrieved sentence ∩relevant sentence|		Precision		Recall		F-measure	
	Our approach	Graph theory	Our approach	Graph theory	Our approach	Graph theory	Our approach	Graph theory	Our approach	Graph theory	Our approach	Graph theory
D1	4	4	8	8	4	2	1	0.5	0.5	0.4	0.667	0.444
D2	3	4	8	8	2	5	0.666	0.75	0.25	0.6	0.364	0.666
D3	5	5	9	9	4	5	0.8	0.75	0.44	0.6	0.568	0.666
D4	5	5	8	9	5	4	1	0.6	0.625	0.6	0.769	0.6