Journal of Computer Science 6 (11): 1366-1376, 2010
ISSN 1549-3636


# Implementation and Evaluation of Evolutionary Connectionist Approaches to Automated Text Summarization

[1]Rajesh Shardanand Prasad and [2]Uday Kulkarni
[1]Department of Computer Engineering, VIIT, Kondhwa, Pune-411048, India
[2]Department of Computer Engineering, SGGS, Vishnupuri, Nanded, India

**Abstract: Problem statement:** Text summarization takes care of choosing the most significant portions of text and generates coherent summaries that express the main intent of the given document. This study aims to compare the performances of the three text summarization systems developed by the authors with some of the existing Summarization systems available. These three approaches to text summarization are based on semantic nets, fuzzy logic and evolutionary programming respectively. All the three represent approaches to achieve connectionism. **Approach:** First approach performs Part of Speech (POS) tagging, semantic and pragmatic analysis and cohesion. The second system under discussion was a new extraction based automated system for text summarization using a decision module that employs fuzzy concepts. Third system under consideration was based on a combination of evolutionary, fuzzy and connectionist techniques. **Results:** Semantic net approach performs better than the MS Word summarizer as far as the semantics of the original text was concerned. To compare our summaries with those of the well known MS Word, Intellexer and Copernic summarizers, we use DUC's human generated summaries as the bench-mark. The results were very encouraging. The second approach based on fuzzy logic results in an efficient system since fuzzy logic mimics decision making of humans. Third system showed promising results as far as precision and F-measure are concerned than all the other approaches. **Conclusion:** Our first approach used WordNet, a lexical database for English. Unlike other dictionaries, WordNet does not include information about etymology, pronunciation and the forms of irregular verbs and contains only limited information about usage. To overcome this limitation, we developed a new text summarizer based on fuzzy logic. As Text summarization application requires learning ability based on activation, we utilize ANN attribute through a connectionist model to achieve the best results.

**Key words:** Neural network, feature extraction, text summarization, part of speech, evolutionary connectionist, semantic net, perceptron neural network, evolutionary programming, chromosomes, automatic text, semantic nets

## INTRODUCTION

Connectionism is a technical term for a group of related techniques. These techniques include areas such as Artificial Neural Networks, Semantic Networks and a few other similar ideas.

Over the past half a century, the problem of text summarization has been addressed from many different perspective, in various domains and using various paradigms. This study intends to investigate Connectionist architecture for the Text Summarization system, taking into account of existing new developments in adaptive evolving systems. Evolving processes, through both individual development and evolution, inexorably led the human race to our supreme intelligence and our superior position in the animal kingdom.

In this study, we consider the system of an Automatic Text Summarization as Evolving system which learns incrementally through experience in the environment. This study highlights the practical experiences, Connectionist learning environment and new ideas to promote further validations.

**Practical experiences:** Despite the successfully developed and used methods of Computational Intelligence (CI), such as Artificial Neural Networks (ANN), Fuzzy Systems (FS), evolutionary computation, hybrid systems and other methods and techniques, there

**Corresponding Author:** Rajesh Shardanand Prasad, Department of Computer Engineering, Vishwakarma Institute of Information Technology, Pune, 411048, India

are a number of problems while applying these techniques to Text Summarization problem:

- Difficulty in preselecting the system's architecture
- Catastrophic forgetting
- Excessive training time required
- Lack of knowledge representation facilities

To overcome the above problems, improved and new connectionist and hybrid methods and techniques are required both in terms of learning algorithms and systems learning (Richard *et al*., 2008; AL-Salami, 2009; Boukerram and Azzou, 2006; Hergli *et al*., 2005).

**Connectionist learning environment:** An Evolving Connectionist System is an adaptive, incremental learning and knowledge representation system that evolves its structure and functionality (Zhijun and Minghong, 2005). Evolving Connectionist System is a CI system based on neural networks, but using other techniques of CI that operate continuously in time and adapt their structure and functionality through a continuous interaction with the environment. Figure 1 explains a typical connectionist learning environment.

This study describes three approaches to Automated Text Summarization using connectionist statures based on:

- word net, an online dictionary based on Semantic Nets (SN)
- Fuzzy logic
- Evolutionary connectionist and fuzzy techniques

This study is organized as follows: Introduction to the domain immediately follows details of implementation details of all the three summarizers specified above. This is followed by Results, Discussion and Acknowledgement.

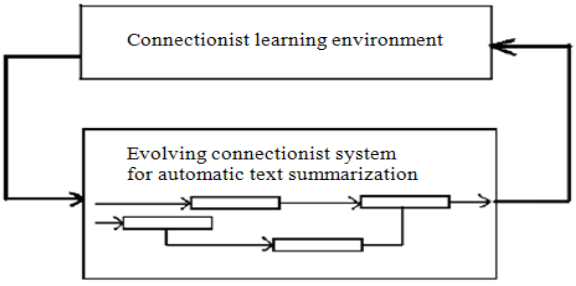


Fig. 1: Evolving connectionist systems evolve their structure and functionality through incremental learning in time and interaction with the environment

Authors have surveyed the current text summarization approaches (Afantenos *et al*., 2005; Ledeneva *et al*., 2008) their advantages and disadvantages and, with the goal of identifying summarization techniques most suitable to generic text summarization. Precision/recall schemes, as well as summary accuracy measures which incorporate weightings based on multiple human decisions, are suggested as particularly suitable in evaluating generic summaries.

## MATERIALS AND METHODS

**Text summarization based on word net, an online Dictionary based on Semantic Nets (SN):** Methods for text classification and information retrieval have been recently presented making use of the word net ontology. Generally, this methodology requires statistical induction of synset clusters and entails costly training of specific key domains. The present study word net is rich enough to obtain useful results in text categorization and summarization without training the tagged corpora.

**Part of Speech (POS) tag and dependency tree generation:** We use the Stanford POS tagger to identify nouns and adjectives in the sentences as shown in Fig. 2. The Stanford POS tagger tags Nouns and Noun Phrases as NN, NNP, NNS and adjectives as JJ. Furthermore, a sentence could contain more than one Noun or Noun Phrases (features) and Adjectives (opinions). Thus, we need to determine the Noun that a particular Adjective modifies i.e. the feature about which a certain opinion has been expressed. For this purpose we used the Stanford Parser to generate the parse tree of a sentence and extract typed dependencies among the words of a sentence.

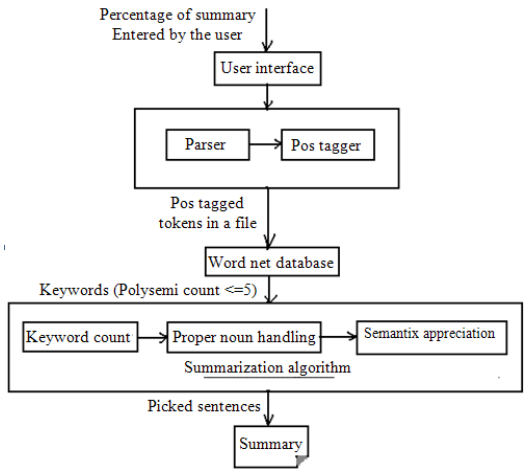


Fig. 2: System architecture of text summarizer based on graph theory

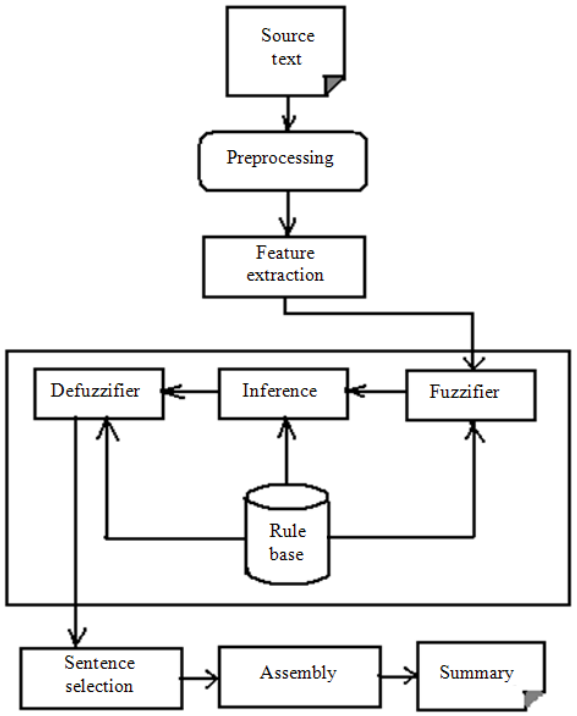


Fig. 3: System architecture of text summarizer based on fuzzy logic

The typed dependencies provide a description of grammatical relationships between the words of sentence.

**Summarization algorithm module:** After the semantic grading of the nouns and verbs, also called nuclei, has been done, keywords among the nuclei are identified these keywords are nuclei having a semantic grading or polysemy count of <=5. Also, modifiers i.e., adjectives and adverbs, having a semantic grading of <=5 are picked, given that they relate to keywords. A separate algorithm is developed to determine which modifiers apply to a nucleus. After all the keywords have been determined, keyword counts of each and every sentence, the semantic unit of summary, are determined. Then, the semantic appreciation of each sentence i.e. modifier effect on nuclei is determined. Using these two criteria and considering proper nouns, sentences for the summary is picked.

**Text summarization based on fuzzy logic:** We next focus on the second system that is an automated text summarization system based on statistical approach using fuzzy logic over some significant text features. Significant text features considered in the design of the proposed system are word similarity among sentences, word similarity among paragraphs, iterative query score, format based score, numerical data, cue-phrases, term weight, thematic features and title features. The extracted text features are then mapped into the fuzzy logic to score each sentence. The summary is extracted from the document based upon the score of each sentence. The proposed automated text summarization system consists of five components: Preprocessing, feature extraction, fuzzy logic scoring, sentence selection and assembly. The system architecture is illustrated in Fig. 3.

**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
- Stop words (Pant *et al*., 2004) 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
- Word stemming (Lovins, 1968) 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

**Feature extraction:** The document after preprocessing is subjected to feature extraction by which the properties of the sentences are extracted to score the sentence. The significant text features considered in the proposed system are:

**Word similarity among sentences:** A sentence is given a high score when the terms or the words in it occur in more number of other sentences in the document. Each sentence is segmented into individual words; the segmented words are searched in the other sentences of the given document. The number of other sentences in which a given word has occurred is termed as the occurrence count of the word. The occurrence count of all the individual words in the sentence is summed up to get the Sentence Occurrence Count (SOC). The score for the feature, word similarity among sentences is calculated as the ratio of the

sentence occurrence count of the given sentences to the maximum sentence occurrence count in the document:

$$WWS\_f(s) = \frac{SOS(S)}{\text{maximum SOC in the document}} \quad (1)$$

**Word similarity among paragraphs:** The feature is extracted from the whole paragraph rather than from individual sentences. Thus all the sentences under a single paragraph will get the same score. This feature is analogous to the word similarity among sentences and the Paragraph Occurrence Count (POC) of a given paragraph is defined as the number of paragraphs in the document that contains the same terms or words as the given paragraph:

$$WSP\_f(p) = \frac{POC(P)}{\text{Maximum POC in the document}} \quad (2)$$

**Iterative query score:** The score corresponding to this feature is accomplished by three phases:

- Initial keyword identification: The top 'n' frequent words are selected as initial keyword set
- Scoring sentences based on iterative query

Query is nothing but searching for a keyword in the given document and retrieving those sentences that contains the keyword. A tag named count will be added to all the sentences in the document which keeps track of the number of appearance of the sentences in the query result of all iteration. In every iteration, the tag count of each sentence will be updated. The iteration stops when predefined number of loops is executed or if there is no change in the keyword list. The sentence score for this feature is the ratio of the count to the total number of iterations:

$$IQ\_f(s) = \frac{counts(s)}{\text{Total number of iteration}} \quad (3)$$

where count is the number of iterations in which the sentence has occurred.

**Format based score:** In many of the documents the importance of the sentences or headings is indicated by expressing the text in different text format e.g., Italics, Bold, underlined, big font size and more. This feature is some what specific to a single sentence and do not depend upon the whole document. By considering the format of the words in the text we can assign a score to the sentence:

$$FB\_f(s) = \frac{\text{Number of words in the sentence with special format}}{\text{Total number of words in the sentence}} \quad (4)$$

**Numerical data:** The numerical data in the document generally brings about some important stats of the core idea of the document and thus the sentence with numerical data can reflect the intention of the document and may be selected for the summary. The score for this feature is calculated as the ratio of the number of numerical data that occur in sentence over the sentence length:

$$NU\_f(s) = \frac{\text{Length of numerical data in the sentence}}{\text{Sentence length}} \quad (5)$$

**Cue-phrases:** Generally 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 important content of a document (Zadeh, 1965). The sentences that contain cue words/phrases are given a higher score than those not containing them. If the sentence contains the cue phrases the sentence gains a score calculated by:

$$CP\_f(s) = \frac{\text{No. of cue words in the sentence}}{\text{Total no. of cue phrases in the documents}} \quad (6)$$

**Term weight:** The term weight of all the terms or words in the given document is calculated and stored for all the words. The term weight for each word is given by the following formula:

$$W_i = TF \times ISF \quad (7)$$

where is the frequency of a particular term that appears in the document and is given by:

$$ISF(t) = \log(NS / SF(t)) \quad (8)$$

where, t is a term in the sample document is the total number of sentences in the document and is the number of sentences in which t occurred. The summation of the calculated term weight of all the terms in a sentence gives the sentence weight in the document:

$$W\_S(S) = \sum_{i \in s} W_i \quad (9)$$

Where:
W_S = Sum of term weights in a sentence
n = Number of words in the sentences

$W_i$ = Weight of the ith word in the sentence S

The score of a sentence is calculated as the ratio of the sentence weight to the maximum sentence weight in the given document:

$$TW\_f(s) = \frac{W\_S(S)}{\text{Maximum sentence weight in the given document}} \quad (10)$$

**Thematic features:** Thematic words are the most frequent words in the given document. The number of thematic words indicates the words with maximum possible relativity. The top n frequent content words are considered as thematic words. The score for this feature is calculated by the following formula:

$$TR\_f(s) = \frac{\text{Number of thematic words in S}}{\text{Maximum (No. of thematic words)}} \quad (11)$$

**Title features:** A sentence is given a higher score if it contains the words that occur in the title. The sentence which contains the words that occur in title may give what the document is intended to express. The score of a sentence for this feature can be calculated as the ratio of the number of words in the sentence that occur in title to the total number of words in the title.

$$T\_f(s) = \frac{\text{Number of title words in S}}{\text{Number of words in the title}} \quad (12)$$

Every sentence in the document along with its ID has a feature vector with nine fields for the aforesaid nine features. All the features will have the value range between 0 and 1.

**Fuzzy logic scoring:** Fuzzy logic was introduced by Zadeh in the late 1960s (Zadeh, 1965) and is considered as the rediscovery of multi-valued logic. In fuzzy logic, the truth values of the variables can take any value in the range 0-1 (e.g., 0.23), in contrast to Boolean logic, in which variables can be either 1 or 0. Triangular membership function and fuzzy logic are used to score a sentence based on the above extracted text features. The Fuzzy logic system consists of four parts:

- Fuzzifier
- Rule base
- Inference engine
- Defuzzifier

**Fuzzifier:** A fuzzifier converts the input feature values into linguistic values (Very Low, Low, Medium, High and Very High) using the membership function. The linguistic value denotes a fuzzy set (e.g., Low) to which a given sentence feature belongs. Fuzzy set FS can be defined as set of ordered pair:

$$FS = (x, f(x)) \quad (13)$$

Where

FS-> = {Very Low, Low, Medium, High, Very High} x->[0,1]

x = Denotes a text feature of the given sentence

f(x) = The Triangular membership function of a fuzzy set given by:

$$f(x) = \begin{cases} 0, & \text{if } x < a \\ \frac{x-a}{b-a}, & \text{if } a \le x \le b \\ \frac{c-x}{c-b}, & \text{if } b \le x \le c \end{cases} \quad (14)$$

where a, b, c are characteristic parameters of a fuzzy set S.

The linguistic value of a given sentence can be determined using the support of each fuzzy set. The support (supp) of a fuzzy set FS is nothing but the list of sentences which give non-zero values for the membership function of FS:

$$Supp(FS) = \{x \in X \mid f(x) > 0\} \quad (15)$$

where is the set of sentences in the document?

It is enough to check all the fuzzy set for the given sentence to determine the linguistic value of a sentence. The linguistic value is the name of the fuzzy set in whose support list the given sentence occurs. There is a possibility that a sentence may belong to more than one fuzzy set, in this case the sentence is considered to belong to the fuzzy set whose membership function gives minimum value for the given sentence.

**Rule base:** The most important procedure in any fuzzy system is defining the fuzzy IF-THEN rules. A rule consists of two parts antecedent and consequent. Antecedent is the possible input feature values and consequent is the inference of the rule which determines whether the sentence is important, average or unimportant based on the input. Sample of fuzzy rule is given below:

IF (Word co-occurrence among sentence is H) and (Word co-occurrence among paragraph is VH) and

(Iterative query score is H) and (Format based score is M) and (No. cue-phrases is H) and (Term weight is VH) and (Thematic feature is VH) THEN (Sentence is important)

**Inference engine:** The Inference engine compares the fuzzy input obtained from the fuzzifier with the Knowledge base and decides the importance of a sentence. The output of inference engine is one of the linguistic values from the set {Unimportant, Average and Important}.

**Defuzzifier:** The linguistic values obtained from the inference engine are converted into crisp values by the defuzzifier. The crisp value denotes how close the sentence is to the given linguistic value.

**Sentence selection and assembly:** The selection of a sentence consists of two steps: (1) determining the number of sentence to be in the summary based on compression rate and (2) extracting the appropriate sentences for the summary. The number of sentences N to be placed in the summary is calculated as:

$$\text{No. of sentences the summary (N)} = \frac{\text{Compression rate}}{100} \times \text{Total no. of sentences in the document} \quad (16)$$

Sentence extraction is accomplished by first arranging the sentences in descending order based on the crisp output value from defuzzifier and the top N sentences are selected for the summary. To obtain clear and logical summary, sentences that are selected to be included in the summary are sequentially ordered based on the order of the reference number or unique ID of the sentence.

**Text summarization based on evolutionary connectionist and fuzzy techniques:** In our prior work (Prasad and Kulkarni, 2009a; 2009b; Prasad *et al*., 2009a; 2009b, 2009c), we described an automatic text summarization system using fuzzy logic. Authors aim to introduce an efficient and effective system for automated text summarization that combines evolutionary, connectionist and fuzzy techniques.

Figure 4 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

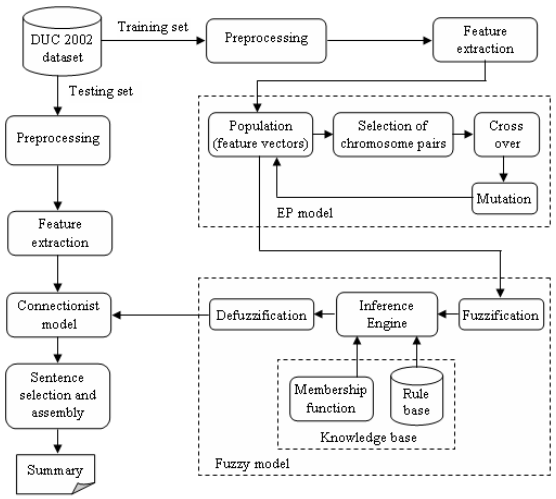


Fig. 4: The proposed automatic text summarization system architecture

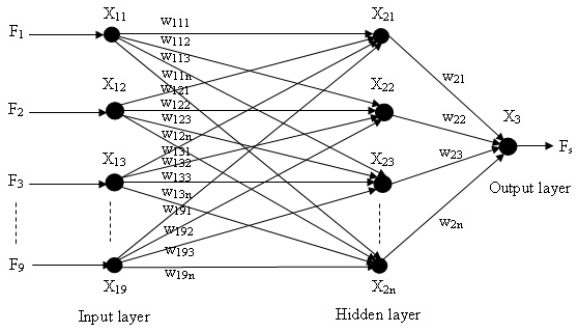


Fig. 5: Structure of multi-layer perceptron neural network

- Fuzzy model
- Evolutionary Programming (EP) model
- Connectionist model
- Sentence selection and assembly

Preprocessing, feature extraction and Fuzzy model are same as explained in later part of this study.

**Evolutionary Programming (EP) model:** The preprocessed sentences is subjected to feature extraction process so that, the feature vector is computed for each sentence. Evolutionary Programming (EP) module generates large number of feature vectors (chromosomes) iteratively utilizing cross over and mutation operators subsequently, fuzzy logic is employed on the chromosomes and it returns

the fuzzy score for each chromosome. Then, the chromosomes with their fuzzy score are fed to the neural network for training.

**Connectionist model:** Authors describe here, the connectionist 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 multilayer perceptron is a feed forward 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) (Lahoz and Miguel, 2006). The neural network used in the proposed system is configured with a nine input, hidden and one output layer. The configurations of the network used for our approach is shown in the Fig. 5.

**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 (Aliruliyev, 2009). 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 (Kursk *et al*., 2006).

Testing phase-In testing phase, 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 input, we make use of EP model, which is based on the genetic operators such as, cross over and mutation.

The first step is the generation of an initial population for evolutionary process. The feature vector of a sentence is known as chromosome (candidate). The set of chromosomes are obtained for every sentence in the text document. Then, by making use of evolutionary concept, more candidates are generated from an initial population. In order to generate large number of candidate sets, we have used the genetic operators such as cross over and mutation (Haupt and Haupt, 1997).

**Selection:** Two random integers are generated within the size of the population. Then, two chromosomes corresponding to the generated number are selected from the initial population.

**Crossover:** The crossover operator is applied on the selected two candidates and this produces two individuals newly. Here, we have used the single point cross over.

**Mutation:** The obtained new set of individuals is then fed to the mutation operator. To have a better exploration of the search space, mutation operator is carried out. Again, we obtain two individuals newly from the single point mutation operator.

**Termination:** The population is updated with four new set of individuals. Again, the selection, crossover and mutation operators are performed iteratively.

**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 to be placed in the summary is calculated as:

$$N = \frac{C \times N_S}{100} \quad (17)$$

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 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.

## RESULTS

The experimental results and analysis of the proposed automatic text summarization system is presented here. The proposed system is implemented in MATLAB (MATLAB 7.8). We have used DUC 2002 dataset in the proposed system for generating the single document summary. DUC 2002 dataset contains documents on different categories and extractive summary per document.

Table 1: Feature score for the text document (document no. AP8803314-0110)

| Sentence ID | Feature score | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | $F_1$ | $F_2$ | $F_3$ | $F_4$ | $F_5$ | $F_6$ | $F_7$ | $F_8$ | $F_9$ |
| $S_1$ | 1.0000 | 1 | 1 | 0.1739 | 0.0000 | 0.3596 | 1.0 | 1.00 | 0.2739 |
| $S_2$ | 0.8571 | 1 | 1 | 0.0417 | 0.0121 | 0.2895 | 0.8 | 0.50 | 0.0000 |
| $S_3$ | 0.5714 | 1 | 1 | 0.0000 | 0.0000 | 0.2982 | 0.4 | 0.00 | 0.0000 |
| $S_4$ | 0.2857 | 1 | 1 | 0.0000 | 0.0000 | 0.2895 | 0.2 | 0.00 | 0.0000 |
| $S_5$ | 0.5714 | 1 | 1 | 0.0000 | 0.0000 | 0.3158 | 0.6 | 0.25 | 1.0000 |
| $S_6$ | 0.5714 | 1 | 1 | 0.0000 | 0.0000 | 0.3070 | 0.6 | 0.25 | 0.0000 |
| $S_7$ | 0.7143 | 1 | 1 | 0.0000 | 0.0000 | 0.3509 | 0.4 | 0.75 | 0.0000 |
| $S_8$ | 0.5714 | 1 | 1 | 0.0000 | 0.0000 | 0.3509 | 0.6 | 0.50 | 0.0000 |

Table 2: Feature score for the text document (document no. AP8803314-0110)

| Sentence ID | Fuzzy score |
|---|---|
| $S_1$ | 0.5095 |
| $S_2$ | 0.5078 |
| $S_3$ | 0.5082 |
| $S_4$ | 0.5178 |
| $S_5$ | 0.5082 |
| $S_6$ | 0.5082 |
| $S_7$ | 0.5086 |
| $S_8$ | 0.5078 |

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.

Then, we apply genetic operators on the 100 feature vectors in order to attain the 2000 feature vector. 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.

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 preprocessing 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.

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.

**Evaluation measure:** The performance of the proposed approach is evaluated using precision, recall and F-measure. 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:

$$\text{Precision} = \frac{|\{\text{Retrieved sentences}\} \cap \{\text{Relevant sentences}\}|}{|\{\text{Retrieved sentences}\}|} \quad (18)$$

$$\text{Recall} = \frac{|\{\text{Retrived sentences}\} \cap \{\text{Relevant sentences}\}|}{|\{\text{Relevant sentences}\}|} \quad (19)$$

Where:

Relevant sentences = Sentences that are identified in the human generated summary

Retrieved sentences = Sentences that are retrieved by the system

$$\text{F-measure} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (20)$$

**Performance evaluation:** The performance of the proposed system is evaluated on the summary available in the DUC 2002 dataset using the evaluation measures described above. We have taken four documents from the dataset, $D_1$ (AP880310-0062), $D_2$ (AP880622-0184), $D_3$ (AP880816-0135) and $D_4$ (FT923-5835). Then, we generate the single document summary for these four documents using the proposed system. For experimentation, 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.

Table 3: Feature score for the text document (document no. LA080890-0078)

| | Feature score | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Sentence ID | $F_1$ | $F_2$ | $F_3$ | $F_4$ | $F_5$ | $F_6$ | $F_7$ | $F_8$ | $F_9$ |
| $S_1$ | 0.8571 | 1 | 1 | 0.4 | 0 | 0.3070 | 1.0000 | 1 | 0.0000 |
| $S_2$ | 0.8571 | 1 | 1 | 0.0 | 0 | 0.3509 | 0.6667 | 0 | 0.0000 |
| $S_3$ | 0.8571 | 1 | 1 | 0.0 | 0 | 0.3684 | 0.5000 | 0 | 1.0000 |
| $S_4$ | 1.0000 | 1 | 1 | 0.0 | 0 | 0.3333 | 0.6667 | 0 | 0.0000 |
| $S_5$ | 0.8571 | 1 | 1 | 0.0 | 0 | 0.3684 | 0.3333 | 0 | 0.0000 |
| $S_6$ | 0.8571 | 1 | 1 | 0.0 | 0 | 0.3509 | 0.6667 | 0 | 0.0000 |
| $S_7$ | 0.1429 | 1 | 0 | 0.0 | 0 | 0.3509 | 0.0000 | 0 | 0.6647 |
| $S_8$ | 0.8571 | 1 | 1 | 0.0 | 0 | 0.3596 | 0.5000 | 0 | 0.5775 |

Table 4: Feature score for the text document (document no. LA080890-0078)

| Sentence ID | Sentence score |
|---|---|
| $S_1$ | 0.6129 |
| $S_2$ | 0.6108 |
| $S_3$ | 0.5138 |
| $S_4$ | 0.5876 |
| $S_5$ | 0.5862 |
| $S_6$ | 0.6108 |
| $S_7$ | 0.5228 |
| $S_8$ | 0.5597 |

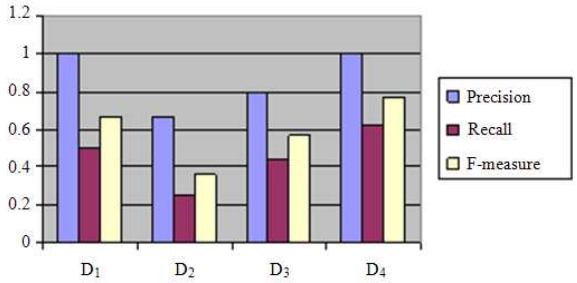


Fig. 6: Comparison graph for compression rate C = 40

The computed evaluation measures for compression rate C = 40 is given in Table 5 and their corresponding graph is shown in Fig. 6. Similarly, the performance graph is plotted for compression rate C = 50 and C = 60, which is shown in Table 6 and 7 and Fig. 7 and 8.

**Performance comparison with other methods:** To test the summarization process we initially summarized different articles on variety of domains, such as politics, literature, spirituality, sports and technology. Sub domains such as comic, fiction, news articles, children stories were also included in literature category. The purpose was to test the context understanding by the summarizers developed by us.

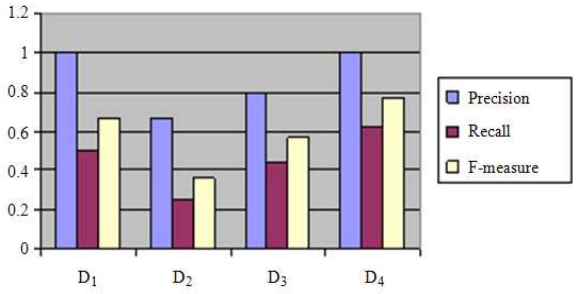


Fig. 7: Comparison graph for compression rate C = 50

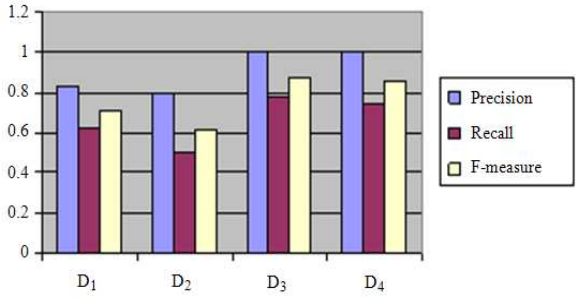


Fig. 8: Comparison graph for compression rate C = 60

The authors compare the average precision, recall and F-measure score between Copernic, Intellexer, General Statistic Method (GSM), Microsoft Word 2007 summarizer systems and the three summarizers developed by the authors (Prasad and Kulkarni, 2009a; 2009b; Prasad *et al*., 2009a; 2009c, 2009c). The results shown in Table 8 show Approach one reaches the average precision of 0.70000, recall of 0.76666 and F-measure of 0.65555, the second summarizer achieves the average precision of 0.83051, recall of 0.79000 and F-measure of 0.83 and connectionist summarizer achieves precision of 1, recall 0.77 and F-measure of 0.87.

Table 5: Precision, recall and F-measure for comparison rate C = 40

| | Retrieved sentences | Relevant sentences | Retrieved sentences ?Relevant sentences | Precision | Recall | F-measure |
|---|---|---|---|---|---|---|
| $D_1$ | 4 | 8 | 4 | 1.0000 | 0.500 | 0.666666667 |
| $D_2$ | 3 | 8 | 2 | 0.6666 | 0.250 | 0.363626446 |
| $D_3$ | 5 | 9 | 4 | 0.8000 | 0.440 | 0.567741935 |
| $D_4$ | 5 | 8 | 5 | 1.0000 | 0.625 | 0.769230769 |

Table 6: Precision, recall and F-measure for comparison rate C = 50

| | Retrieved sentences | Relevant sentences | Retrieved sentences? Relevant sentences | Precision | Recall | F-measure |
|---|---|---|---|---|---|---|
| $D_1$ | 5 | 8 | 5 | 1.000 | 0.625 | 0.769230769 |
| $D_2$ | 4 | 8 | 3 | 0.750 | 0.375 | 0.500000000 |
| $D_3$ | 6 | 9 | 5 | 0.833 | 0.555 | 0.666159940 |
| $D_4$ | 6 | 8 | 6 | 1.000 | 0.750 | 0.857142857 |

Table 7: Precision, recall and F-measure for comparison rate C = 60

| | Retrieved sentences | Relevant sentences | Retrieved sentences? Relevant sentences | Precision | Recall | F-measure |
|---|---|---|---|---|---|---|
| $D_1$ | 6 | 8 | 5 | 0.833 | 0.6250 | 0.714163237 |
| $D_2$ | 5 | 8 | 4 | 0.800 | 0.5000 | 0.615384615 |
| $D_3$ | 7 | 9 | 7 | 1.000 | 0.7777 | 0.874950779 |
| $D_4$ | 6 | 8 | 6 | 1.000 | 0.7500 | 0.857142857 |

Table 8: Comparison of the three summarizers with some well known summarizers

| Summarizers | Precision | Recall | F-measure |
|---|---|---|---|
| Copernic | 0.8000 | 0.7750 | 0.78600 |
| Intellexer | 0.8250 | 0.7083 | 0.75590 |
| MS word | 0.5916 | 0.6250 | 0.59130 |
| GSM | 0.4904 | 0.4356 | 0.45542 |
| Approach 1: Semantic nets | 0.7666 | 0.6555 | 0.70000 |
| Approach 2: Fuzzy LOGIC | 0.7900 | 0.7900 | 0.83050 |
| Approach 3: Connectionist | 1.0000 | 0.7700 | 0.87000 |

## DISSCUSSION

We have developed automatic text summarization system with three different approaches. The purpose was to implement and evaluate existing connectionist methods and adopt the best suited for the domain of text summarization process. The experimental results show that the third approach, which combines EP model, Artificial Neural Network (ANN) and fuzzy logic suits the said domain appropriately. Here, we have used nine different features for feature extraction phase. Then, the feature vectors are iteratively generated by making use of EP model. Subsequently, 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.

## CONCLUSION

Since a lot of interesting work is being done far from the mainstream research in this field, we have chosen to develop approaches to Text Summarization that we found relevant to future research, even if they focus only on small details related to a general summarization process and not on building an entire summarization system. The results obtained, suggest that the future of this research area heavily depends on the ability to find efficient ways of automatically evaluating these systems and on the development of measures that are objective enough to be commonly accepted by the research community.

## ACKNOWLEDGMENT

This project is in process of funding to be granted by Board of College and Universities Development BCUD under University of Pune.

## REFERENCES

Afantenos, S., V. Karkaletsis and P. Stamatopoulos, 2005. Summarization from medical documents: A survey. Artif. Intell. Med., 33: 157-177. DOI: 10.1016/j.artmed.2004.07.017

Aliruliyev, R.M., 2009. A new sentence similarity measure and sentence based extractive technique for automatic text summarization. Exp. Syst. Appli. 36: 7764-7772. DOI: 10.1016/j.eswa.2008.11.022

AL-Salami, N.M.A., 2009. Evolutionary algorithm definition. Am. J. Eng. Applied Sci., 2: 789-795. DOI: 10.3844/.2009.789.795

Boukerram, A. and S.A.K. Azzou, 2006. Implementation of load balancing algorithm in a grid computing. Am. J. Applied Sci., 3: 1810-1813. DOI: 10.3844/.2006.1810.1813

El Emary, I.M.M. and F. Al Taweel, 2005. An advance approach to evaluate the performance of the TCP networks. Am. J. Applied Sci., 2: 1375-1379. DOI: 10.3844/.2005.1375.1379

Haupt, R.L. and S.E. Haupt, 1997. Practical Genetic Algorithms. 1st Edn., Wiley-Interscience, New York, ISBN: 10: 0471188735, pp: 192.

Hergli, M., J. Baili, F. Bouslama and K. Besbes, 2005. A new compressing ultrasonic data algorithm based on wavelets. Am. J. Applied Sci., 2: 1615-1618. DOI: 10.3844/.2005.1615.1618

Kursk, S.M., R.J. Rasras and D. Skopin, 2006. The artificial neural network based approach for mortality structure analysis. Am. J. Applied Sci., 3: 1698-1702. DOI: 10.3844/.2006.1698.1702

Lahoz, D. and M.S. Miguel, 2006. MLP neural network to predict the wind speed and direction at Zaragoza. Monografías del Seminario Matemático García de Galdeano, 33: 293-300. http://www.unizar.es/galdeano/actas_pau/PDFIX/LahSan05.pdf

Ledeneva, Y., A. Gelbukh and R.A. Garcia-Hernandez, 2008. Terms derived from frequent sequences for extractive text summarization. Proceedings of the 9th International Conference on Computational Linguistics and Intelligent Text, Feb. 17-23, Springer-Verlag Berlin, Haifa, Israel, pp: 593-604. http://portal.acm.org/citation.cfm?id=1787643

Lovins, J.B., 1968: Development of a stemming algorithm. Mech. Trans. Comput. Linguist., 11: 22-31. DOI: 10.1234/12345678

Pant, G., P. Srinivasan and F. Menczer, 2004. Crawling the Web. In: Web Dynamics: Adapting to Change in Content, Size, Topology and Use, Levene, M. and A. Poulovassilis (Eds.). Springer, USA., pp: 153-178.

Prasad, R.S. and U.V. Kulkarni, 2009a. An automated extraction based system for effective summarization of text documents using fuzzy logic. Int. J. Comput. Eng. Inform. Technol. (In Press).

Prasad, R.S. and U.V. Kulkarni, 2009b. Two approaches to automatic text summarization: Extractive methods and evaluation. Int. J. Comput. Eng. Comput. Appli., 1: 24-36.

Prasad, R.S., U.V. Kulkarni and J.R. Prasad, 2009a. Machine learning in evolving connectionist text summarizer. Proceedings of IEEE International Conference on Anti-Counterfeiting, Security and Identification, Ang. 20-22, IEEE Xplore Press, Hong Kong, pp: 539-543. DOI: 10.1109/ICASID.2009.5277001

Prasad, R.S., U.V. Kulkarni and J.R. Prasad, 2009b. A novel Evolutionary Connectionist Text Summarizer (ECTS). Proceedings of IEEE International Conference on Anti-Counterfeiting, Security and Identification, Ang. 20-22, IEEE Xplore Press, Hong Kong, pp: 606-610. DOI: 10.1109/ICASID.2009.5277003

Prasad, R.S., U.V. Kulkarni and J.R. Prasad, 2009c. Connectionist approach to generic text summarization. J. World Acad. Sci. Eng. Technol., 55: 365-369. http://www.waset.org/journals/waset/v55/v55-63.pdf

Richard, R.J.A., A.A. Joshi and C. Eswaran, 2008. Implementation of computational grid services in enterprise grid environments. Am. J. Applied Sci., 5: 1442-1447. DOI: 10.3844/ajassp.2008.1442.1447

Zadeh, L.A., 1965. Fuzzy sets. Inform. Control, 8: 338-353. DOI: 10.1016/j.fss.2004.03.027

Zhijun, L.I. and L. Minghong, 2005. EOS: Evolutionary overlay service in peer-to-peer systems. Am. J. Applied Sci., 2: 1401-1406. DOI: 10.3844/.2005.1401.1406