

## DOCUMENT SUMMARIZATION USING DIFFERENTIAL EVOLUTION ALGORITHM

S. K. Nayak\*

Y. Karali\*\*

R. Routray\*\*\*

### ABSTRACT

The use of document summarization allows a user to get a sense of the content of full document, or to know its information content without reading all sentences within the document. Data reduction helps user to find the required information quickly without rendering more time in reading the whole document. This paper presents a method to generate a summary from the original document. And the method includes several characteristics such as sentence-id, position of each term in a sentence, term frequency, sentence similarity measure and weight of each and every sentence. To solve the optimization problem differential evolution (DE) algorithm is used, which can choose the optimal summary. DE algorithm is based on a fitness function and selection of fitness function is crucial for the good performance of DE algorithm.

**Keywords:** Document Summarization, DE Algorithm, TF-IDF Method, Vector Space Model.

\* M.Phil Student, PG. Dept. of Comp. Sc. and Applications, Sambalpur University, Odisha, India

\*\* PhD Scholar, PG. Dept. of Comp. Sc. and Applications, Sambalpur University, Odisha, India

\*\*\* Asst. Professor, Dept. of Comp. Sc. and Engg., S O A University, Odisha, India

## 1. Introduction

Document summarization is the process of reducing a text document with a computer program in order to create a summary that retains the most important points of the original document. Document summarization has attracted much attention since the problem of information overload has grown, and the quantity of data has increased.

When human produce summaries of document, they don't simply extract sentences and concatenate them. Rather, they create new sentences that are grammatical, that cohere with one another, and that capture the most salient pieces of information in this original document. Sentence compression is a big challenge in case of summary generation [16].

Generally, there are two types of summarization: extraction and abstraction. Extractive methods work by selecting a subset of existing words, phrases, or sentences in the original text to form the summary. In contrast, abstractive methods build an internal semantic representation and then use natural language generation techniques to create a summary that is closer to what a human might generate [14].

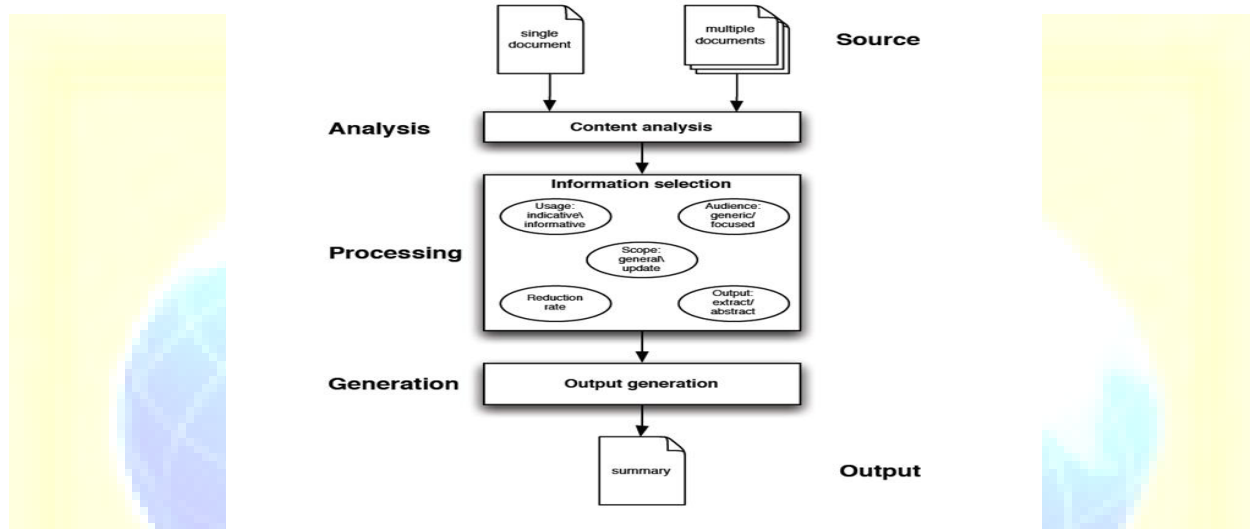
Generally there are two approaches of summarization.

- **Single document summarization:** A set of features is computed for each passage, and ultimately these features are normalized and summed. The passages with the highest resulting scores are sorted and returned as the extract.
- **Multi-document summarization:** Multi-document summarization aims at extracting the major information from multiple documents and has become an important aspect of IR [6,7]. Multi-document summaries are built using a meta summarization procedure. First, for each document in a given cluster of documents, a single document summary is generated using one of the graph-based ranking algorithms. Next, a summary of summaries is produced using the same or a different ranking algorithm [15].

## 2. Related Works

Now-a- days the internet users increase rapidly and as a result the information contributed to internet also increases. The number of websites has grown from 1 million in 1996 to over 1 billion in 2014. This development is leading to information overload. To avoid drowning in information, the flow needs to be filtered and the content condensed. Document summarization can help by providing shortened versions of texts [22].

In general every text summarization system involves three basic steps - analysis, processing and generation. In the first step the document(s) to be summarized are analyzed, e.g., redundant information is identified. In the next step, processing, the information for the summary is selected, for example the redundant information are selected. During generation the actual text of the summary is generated. Although all summarization systems have these three stages in common, different systems produce different summaries [13].



(Figure 1 Basic steps of Text Summarization)

Yan-Xiang He et al. [1] proposed multi-document summarizer using genetic algorithm based sentence extraction (MSBGA) regards summarization process as an optimization problem where the optimal summary is chosen among a set of summaries formed by the conjunction of the original articles sentences. To solve the NP hard optimization problem, MSBGA adopts genetic algorithm, which can choose the optimal summary on global aspect. The evaluation function employs four features according to the criteria of a good summary: satisfied length, high coverage, high informativeness and low redundancy. To improve the accuracy of term frequency, MSBGA employs a novel method TFS, which takes word sense into account while calculating term frequency.

Xiaojuan Zhao et al. [2] proposed a new method of query-focused multi-documents summarization based on genetic algorithm, genetic algorithm is used to extract the sentences to form a summary, and it is based on a fitness function formed by three factors. The proposed summarization method can improve the performance of summary.

Cristina Lopez-Pujalte et al. [3] proposed Order-Based Fitness Function for Genetic Algorithms Applied to Relevance Feedback. Recently there have been appearing new applications of genetic algorithms to information retrieval, most of them specifically to relevance feedback. The evolution of the possible solutions is guided by fitness functions that are designed as measures of the goodness of the solutions. These functions are naturally the key to achieving a reasonable improvement, and which function is chosen most distinguishes one experiment from another.

You Ouyang et al. [4] proposed a study on position information in document summarization. Position information has been proved to be very effective in document summarization, especially in generic summarization. Existing approaches mostly consider the information of sentence positions in a document, based on a sentence position hypothesis that the importance of a sentence decreases with its distance from the beginning of the document.

Mitra M. et al. [17] proposed multi-document summarization by sentence extraction that builds on single document summarization methods by using additional, available information about the document set as a whole and relationships between the documents.

R. Mihalcea et al. [20] proposed a method for automatic book summarization. Most of the text summarization research carried out till date has been concerned with the summarization of short document but this system describes the problem of book summarization.

Zhanying He et al. [21] Document summarization is of great value to many real world applications, such as snippets generation for search results and news headlines generation. Traditionally, document summarization is implemented by extracting sentences that cover the main topics of a document with a minimum redundancy.

### 3. Basic Concepts of Document Summarization

Document summarization is the creation of a shortened version of a text by a computer program. The product of this procedure still contains the most important points of the original text. Document summarization generally occurs in 3 steps.

- (i.) Pre-processing      (ii.) Sentence Extraction      (iii.) post-processing

#### 3.1. Pre-processing

The steps for pre-processing procedure are as follows.

### 3.1.1. Lexical Analysis

The main objective of lexical analysis process is the identification of words in a text. Usually the following cases are considered: digits, hyphens, punctuation marks and the case of the letters (low and upper case).

### 3.1.2. Elimination of Stopwords

A stopword can be a word without meaning in a specific language, or it can be token that does not have linguistic meaning. The examples of stopwords in the English language are "a", "the", "is", etc. Elimination of stopwords has an additional important benefit. It reduces number of terms considerably and generally used for sentence compression [23].

### 3.1.3. Stemming

A stem is the portion of a word which is left after the removal of its affixes. A typical example of a stem is the word "detect" which is the stem of the variants "detected", "detecting", "detection", and "detections". Stems are thought to be useful for improving searching of terms because they reduce variants of the same root word to a common concept.

## 3.2. Sentence Extraction

An evolutionary algorithm (EA) is a subset of evolutionary computation, a generic population-based metaheuristic optimization algorithm.

### 3.2.1. Implementation of Evolutionary Algorithm

1. Generate the initial population of individuals randomly - First Generation
2. Evaluate the fitness of each individual in that population
3. Repeat on this generation until termination (time limit, sufficient fitness achieved, etc)
  - i) Select the best-fit individuals for reproduction – parents.
  - ii) Breed new individuals through crossover and mutation operations to give birth to offspring.
  - iii) Evaluate the individual fitness of new individuals
  - iv) Replace least-fit population with new individuals

One of the evolutionary algorithm (like Genetic algorithm (GA), Differential evolution (DE) algorithm, etc) is used as optimization algorithm to extract the summary sentences from original document. In this proposed system we are using DE algorithm, which can give the optimal summary.

### 3.3. Post Processing

The output from the Sentence Extraction component is a ranked set of sentences selected by the DE algorithm. Again to reduce number of sentences we are taking a threshold value. If length of the sentence is less than the threshold value should be deleted from the summary and the sentences having length more than threshold value should remain in the final summary.

## 4. Vector Space Model

For text representation vector space model is used. To assess the similarity or dissimilarity of two or more documents, we need a model in which these operations are defined. The model is usually selected to match a particular task's requirements and objectives. To keep this chapter's size reasonable we will focus only on Vector Space Model (vsm) and the elements it consists of: document indexing, feature weighting and similarity coefficients.

### 4.1. Document Indexing

Vector Space Model uses the concepts of linear algebra to address the problem of representing and comparing textual data. A document  $d$  is represented in the vsm as a document vector  $[w_{t0}, w_{t1}, \dots, w_{t\omega}]$ , where  $t0, t1, \dots, t\omega$  is a set of words of a given language and  $w_{ti}$  expresses the weight (importance) of term  $t_i$  to document  $d$ . Weights in a document vector typically reflect the distribution of words in that document. In other words, the value  $w_{ti}$  in a document vector  $d$  represents the importance of word  $t_i$  to that document.

Given a set of sentences in a single document (Table-I), their sentence vectors can be put together to form a matrix called a term-frequency matrix.

**Table-I Sentence of a Document**

Sentence	Content
1	Large Scale Singular Value Computations
2	Software for the Sparse Singular Value Decomposition
3	Introduction to Modern Information Retrieval
4	Linear Algebra for Intelligent Information Retrieval
5	Matrix Computations
6	Singular Value Analysis of Cryptograms



Table-II Term Frequency Matrix

Sentence-id	Information	Scale	Analysis	Singular	Value	...
1	0	1	0	1	1	
2	0	0	0	1	1	
3	1	0	0	0	0	...
4	1	0	0	0	0	
5	0	0	0	0	0	
6	0	0	0	0	1	

In the first step, we identify all possible terms appearing in the input and build a matrix where columns correspond to terms and rows correspond to the number of sentences. We exclude certain terms that we know are not useful for identifying the topic of a document (these are called stop words) and restrict the presentation to just a few selected terms with at least one non-zero weight. On the intersection of each column and each row we place the count (number of occurrences) of the column's term in the row's document. For our example input, the term-document matrix looks as shown in Table-II.

#### 4.2. Feature Weighting

Feature weighting methods can be divided into local (one document's term count is available) and global (term counts of all documents are available). Some of the weighting schemes are given below.

i) Term frequency, Inverse document frequency(tf-idf):

Certainly the most widely known feature weighting formula, usually abbreviated to an acronym tf-idf. Credited to Gerard Salton tf-idf tries to balance the importance of a word in a document with how common it is in the entire collection.

ii) Modified tf-idf:

This modification of the original tf-idf downplays the count of terms in a document and contains certain algebraic modifications for faster calculation of  $w(i, j)$  on a preched index.

iii) Pointwise mutual information:

A widely used weighting scheme, although known to be biased towards in- frequent events (terms).

iv) Discounted mutual information:

Similar to Pointwise mutual information, but multiplied with a discounting factor.

#### 4.3. Similarity Co-efficient

Two documents in the Vector Space Model represent two points in a multidimensional term space (each term is assumed to be an independent dimension). If we define a notion of distance in this space, we can compare documents against each other and thus start looking for similarities or dissimilarities.

We can define the cosine measure of similarity between vector representation of documents  $S_i$  and  $S_j$  in the term vector space as:

$$Sim(S_i, S_j) = \cos(\alpha) = \frac{(S_i \cdot S_j)}{|S_i| |S_j|}$$

Where  $x \cdot y$  denotes the dot product between vectors  $x$  and  $y$  and  $|x|$  is the norm of vector  $x$ .

#### 4.4. TF-IDF Method

Tf-idf, term frequency-inverse document frequency, is a numerical statistic which reflects how important a word is to a document in a collection or corpus.

The following notation is used:  $tf_{ij}$  - number of occurrences of term  $i$  in document  $j$ ,  $df_i$  - number of documents containing term  $i$  in the entire collection,  $w(i, j)$  - weight of term  $i$  in document  $j$ ,  $N$  is the number of all documents in the collection.

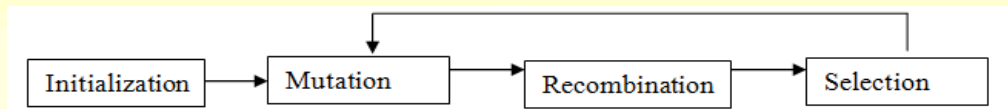
$$w(i, j) = tf_{ij} \times \log_2\left(\frac{N}{df_i}\right)$$

In our proposed system we are generating summary from single document, so we are using only term frequency (tf). Inverse document frequency is generally used for multi-document.



## 5. Differential Evolution Algorithm

Differential Evolution (DE) algorithm is a new heuristic approach [18] mainly having three advantages; finding the true global minimum regardless of the initial parameter values, fast convergence, and using few control parameters. DE algorithm is a population based algorithm like genetic algorithms using similar operators; crossover, mutation and selection. EAs save sufficient data about problem features, search space and population information during runtime [12].



**Figure 2 DE Algorithm Procedure**

### 5.1. Population Initialization

The classical DE is a population based global optimization that uses a real coded representation. Like to other evolutionary algorithms, DE also starts with a population of  $N$  (it must be at least 4). Suppose we want to optimize a function with  $D$  real parameters. The parameter vectors have the form:

$$x_{i,G} = [x_{1,i,G}, x_{2,i,G}, \dots, x_{D,i,G}] \quad i = 1, 2, \dots, N$$

Where  $G$  is the generation number.

Upper and lower bounds for each parameter is defined as:

$$x_j^L \leq x_{j,i,1} \leq x_j^U$$

Randomly select the initial parameter values uniformly in the interval  $[x_j^L, x_j^U]$ . Then mutation and crossover operators are employed to generate new candidate vectors, and a selection scheme is applied to determine whether the offspring or the parent survives to the next generation. The above process is repeated until a termination criterion is reached.

## 5.2. Mutation

Each of the  $N$  parameter vectors undergoes mutation, recombination and selection. Mutation expands the search space. For a given parameter vector  $x_i, G$  randomly select three vectors  $x_{r_1, G}$ ,  $x_{r_2, G}$ ,  $x_{r_3, G}$  such that the indices  $i$ ,  $r_1$ ,  $r_2$  and  $r_3$  are distinct.

Add the weighted difference of two of the vectors to the third:

$$v_{i, G+1} = x_{r_1, G} + F(x_{r_2, G} - x_{r_3, G})$$

$v_{i, G+1}$  is called as the donor vector. Here  $F$  is the mutation factor ranges between  $[0, 1]$ .

## 5.3. Recombination

Recombination incorporates successful solutions from the previous generation. The trial vector  $u_{i, G+1}$  is developed from the elements of the target vector,  $x_i, G$  and the elements of the donor vector  $v_{i, G+1}$ . Elements of donor vector enter the trial vector with probability  $CR$ .

$$u_{j, i, G+1} = \begin{cases} v_{j, i, G+1} & \text{if } rand_{j, i} \leq CR \text{ or } j = I_{rand} \\ x_{j, i, G} & \text{if } rand_{j, i} > CR \text{ or } j \neq I_{rand} \end{cases}$$

$rand_{j, i} \sim U[0, 1]$ ,  $I_{rand}$  is a random integer from  $[1, 2, D]$  and  $I_{rand}$  ensures that  $v_{i, G+1} \neq x_i, G$ .

## 5.4. Selection

The selection operator is described as follows.

The target vector  $x_i, G$  is compared with the trial vector  $v_{i, G+1}$  and the one with the lowest function value is admitted to the next generation.

$$x_{i, G+1} = \begin{cases} u_{i, G+1} & \text{if } f(u_{i, G+1}) \leq f(x_{i, G}) \\ x_{i, G} & \text{Otherwise} \end{cases}$$

Therefore, if the new trial vector yields an equal or higher value of the objective function, it replaces the corresponding target vector in the next generation; otherwise the target vector is retained in the population. Hence, the population either gets better (with respect to the maximization of the objective function) or remains the same in fitness status, but never deteriorates.

Mutation, recombination and selection continue until some stopping criterion is reached.

## 6. System Overview

Our summarization system is design with the extractive framework. Important sentences are extracted and reorganized to form a summary. Thus the whole system is divided into three modules:

- i) Pre-processing    ii) Processing    iii) Summary generation

The flowchart of the system overview is shown in the figure below.

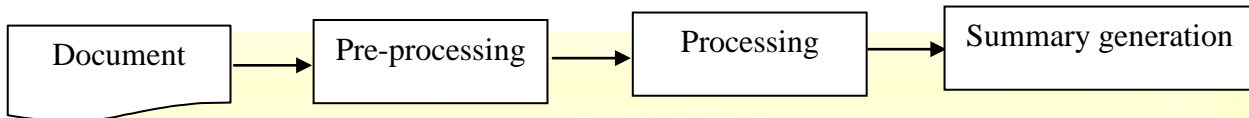


Figure 3 System Flowchart

### 6.1. Pre-processing

A pre-processing procedure is consisting of several steps. In pre-processing, many patterns are used to reduce one or few words from the original sentences without losing much information. As compression rate decreases, the summary will be more concise [8]. We consider following steps for this procedure.

- Sentence Segmentation
- Stopword removal

To remove the stopwords we are using a stopwords list. Figure-4 is showing some of the stopwords.

1	2	3	4	5
1	a			
2	about			
3	above			
4	across			
5	after			
6	afterwards			
7	again			
8	against			
9	all			
10	almost			
11	alone			
12	along			
13	already			
14	also			
15	although			

Figure 4 Stopword list

#### 6.1.1. Sentence Segmentation

First step in summarization process is to segment all the sentences from the document.

### 6.1.2. Stopword removal

First we are extracting the original sentences from the document. Here sentences of the document are modeled as vectors [9] using vector space model. Stopword removal is required to reduce the length of a sentence; as a result it reduces the size of the term-frequency matrix. After optimization removed stopwords are again used with the summary sentences to get the proper meaning of that summary.

The below Figure-5 is shows first ten sentences from the document.

	1	2	3	4	5	6	7	8	9	10
1	'Natural'	'language'	'processing'	'here'	'refers'	'to'	'the'	'use'	'and'	'ability'
2	'The'	'systems'	'of'	'real'	'interest'	'here'	'are'	'digital'	'computers'	'of'
3	'Of'	'course'	'humans'	'can'	'process'	'natural'	'languages,'	'but'	'for'	'us'
4	'First'	'of'	'all,'	'occasionally'	'the'	'phrase'	"'natural'	'language'"	'is'	'used'
5	'Hence'	'one'	'writer'	'states'	'that'	"'human'	'languages'	'allow'	'anomalies'	'that'
6	'I'	'do'	'not'	'use'	'the'	'phrase'	"'natural'	'language'"	'in'	'this'
7	'When'	'I'	'use'	'the'	'phrase,'	'I'	'mean'	'human'	'language'	'in'
8	'There'	'is'	'a'	'broad'	'sense'	'and'	'a'	'narrow'	'sense'	'.The'
9	'At'	'other'	'times'	'the'	'phrase'	'is'	'used'	'more'	'narrowly'	'to'
10	'Even'	'if'	'this'	'confusion'	'is'	'overcome,'	'the'	'phrase'	"'natural'	'language'

Figure 5 Sentences of the document

Stopwords are stored in a table for the future reference. Without stopwords there is no meaning of a sentence. To get the proper meaning of sentences stop- words are reinserted according to sentence-id and position of stopwords. The below Figure-6 shows the sentences after removal of stopwords.

	1	2	3	4	5	6	7	8	9	10
1	'Natural'	'language'	'processing'	'refers'	'use'	'ability'	'systems'	'process'	'sentences'	'natural'
2	'systems'	'real'	'digital'	'computers'	'type'	'think'	'personal'	'computers'	'mainframes'	'(and'
3	'course'	'humans'	'process'	'natural'	'languages'	'question'	'digital'	'computers'	'process'	'natural'
4	'all,'	'occasionally'	'phrase'	'"natural'	'language"'	'used'	'actual'	'languages'	'used'	'ordinary'
5	'writer'	'states'	'"human'	'languages'	'allow'	'anomalies'	'natural'	'languages'	'allow'	'.There'
6	'use'	'phrase'	'"natural'	'language"'	'restricted'	'sense'	'artificial'	'natural'	'language'	[]
7	'use'	'phrase,'	'mean'	'human'	'language'	'messiness'	'varied'	'use'	'The'	'phrase'
8	'broad'	'sense'	'narrow'	'sense'	'.The'	'phrase'	'taken'	'broadly'	'include'	'signal'
9	'times'	'phrase'	'used'	'narrowly'	'include'	'syntactic'	'semantic'	'analysis'	'processing'	[]
10	'confusion'	'overcome,'	'phrase'	'"natural'	'language'	'processing"'	'taken'	'synonymo...	'"natural'	'language'

Figure 6 Sentences without stopwords

Figure-7 is showing some of the stopwords with their sentence-id and position.

swtable.sentenceid	1
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1
11	1
12	1
13	1
14	1
15	1

swtable.term	1
1	here
2	to
3	the
4	and
5	of
6	to
7	in
8	a
9	such
10	as
11	rather
12	than
13	in
14	a
15	computer

swtable.position	1
1	4
2	6
3	7
4	9
5	11
6	13
7	16
8	17
9	20
10	21
11	23
12	24
13	25
14	26
15	29

Figure 7 Stopwords with sentence-id and position

## 6.2. Pre-processing

- Generate term frequency matrix
- Compute similarity measure between sentences

### 6.2.1. Term frequency Matrix

Figure-8 is showing the term-frequency matrix of the original document.

	1	2	3	4	5	6	7	8	9	10
1	1			1	1	1	1	1	1	1
2	0	0	0	0	0	0	0	1	0	0
3	0	0	0	0	0	1	0	2	0	2
4	0	0	0	0	0	1	0	0	0	0
5	0	1	0	0	0	0	0	0	0	3
6	0	1	0	0	0	1	0	0	0	1
7	0	2	1	0	2	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0
10	0	2	1	0	0	0	0	0	0	0
11	0	2	0	0	0	0	0	0	0	3

Figure 8: Term-frequency matrix

### 6.2.2. Sentence Similarity

After sentence segmentation, an effective method is required to compute the similarity between sentences, [5] Sentence similarity is calculated by most widely used vector space model (vsm). In processing procedure we are measuring sentence similarity with the first sentence of the document using cosine similarity. This similarity is based on a threshold value ( $\lambda$ ). If  $S_{im}(S_i, S_j) < \lambda$  then the sentence should be deleted from the document.

Similarity measure plays important roles in information retrieval and Natural language processing [19].

For sentences  $S_i = [P_1, P_2, \dots, P_k]$  and  $S_j = [q_1, q_2, \dots, q_k]$ , the sentence similarity is computed as:

$$Sim(S_i, S_j) = \frac{\sum_k p_i^* q_j}{\sqrt{(\sum_k p_i^2) * (\sum_k q_i^2)}}$$

Figure-9 is showing similarity measure values of first ten sentences.

	1	2	3	4	5	6	7	8	9	10
1	1	0.0396	0.2202	0.0806	0.0389	0.0591	0.3371	0.4264	0.3371	0.34
2										
3										
4										
5										
6										
7										
8										
9										
10										

Figure 9: Sentence similarity measure

### 6.3. Summary generation

In this step Differential evolution algorithm is used to extract the summary sentences. Sentence extraction is an approach to sentence compression [10]. Here sentences of a document are modeled as vectors using vector space model. The execution of the differential evolution is similar to other evolutionary algorithms like genetic algorithms or evolution strategies. The evolutionary algorithms differ mainly in the representation of parameters (usually bi-nary strings



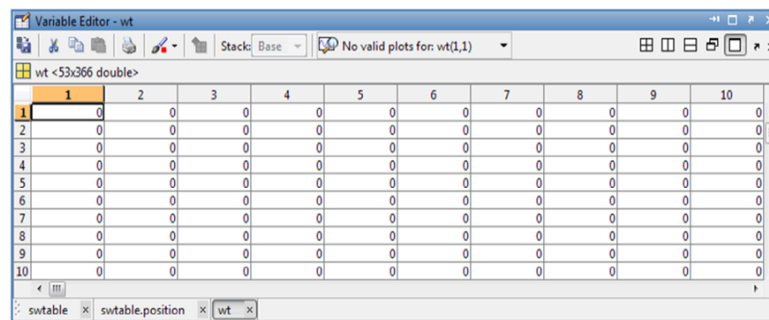
are used for genetic algorithms while parameters are real-valued for evolution strategies and differential evolution) and in the evolutionary operators. To solve the optimization problem differential evolution algorithm is used [11]. After sentence segmentation, an effective method is required to compute the similarity between sentences. Sentence similarity is calculated by most widely used vector space model (vsm). Here we have used Cosine similarity to measure similarity between two vectors of n-dimensions by finding the cosine of the angle between them.

Our aim is to find a summary using DE algorithm. Here in this paper, initial population for DE algorithm is the term frequency matrix. Sentence extraction is an approach to sentence compression. As compression rate decreases, the summary will be more concise. A fitness function (f) is used to calculate the fitness of each chromosome and some control parameters are used like crossover probability (Pc) and mutation probability (Pm).

$$f = \frac{\beta * wt + \gamma * sim}{\beta + \gamma}$$

Where  $\beta$  and  $\gamma$  are real numbers between 0 and 1, defined by the user. The stopping criteria of DE could be a given number of consecutive iterations within which no improvement of summary occurs.

Figure-9 is showing the term-frequency matrix after optimization.



	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0

**Figure 9: Term-frequency matrix after optimization**

Again to reduce the length of the summary we are using a threshold value (depending upon the weight of each summary sentence). If weight of the sentence is less than the threshold value, then it should be deleted from the summary. Final summary sentences are given below in Figure-10.

	1	2	3	4	5	6	7	8	9	10
1	'here'	'to'	'the'	'and'	'of'	'to'	'in'	'a'	'such'	'as'
2	'problem'	'is'	'that'	'the'	'understand...'	'is'	'sometimes'	'in'	'the'	'of'
3	'Allen'	'at'	'first'	'seems'	'to'	'distinguish'	'understand...'	'as'	'a'	'scientific'
4	'discussions'	'of'	'processing'	'by'	'computers,'	'it'	'is'	'just'	'presuppos...'	'that'
5	'the'	'attempt'	'is'	'being'	'made'	'to'	'understand'	'how'	'processing'	'occurs'
6	'usual'	'goal'	'is'	'to'	'the'	'into'	'some'	'sort'	'of'	'knowledge'

Figure 10: Summary sentences

After getting final summary sentences next step is to reinsert the stopwords according to their sentence-id and position. Though it is not giving meaningful sentence we are extracting the original sentences.

To solve the optimization problem Differential evolution algorithm is used here. Figure-11 is showing the graph of term-frequency matrix before optimization. X- Axis and Y-Axis are representing number of iterations and average value of term- frequency matrix respectively. Figure-12 is showing term-frequency matrix after optimization.

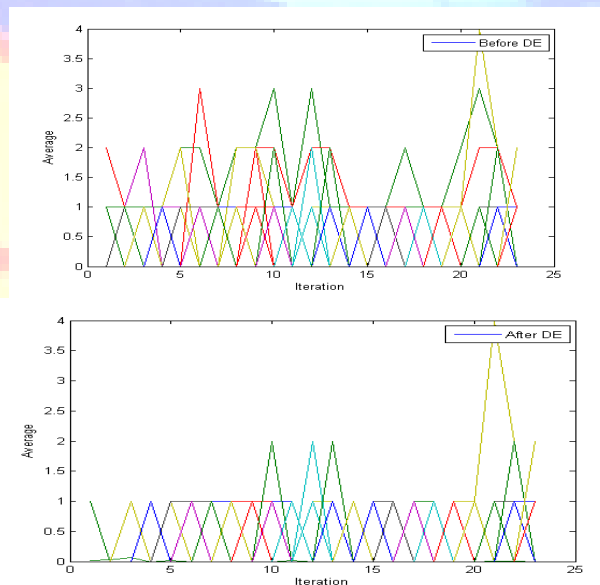


Figure 11: Term-frequency matrix before optimization

Figure 12: Term-frequency matrix after optimization

## 7. Result Analysis

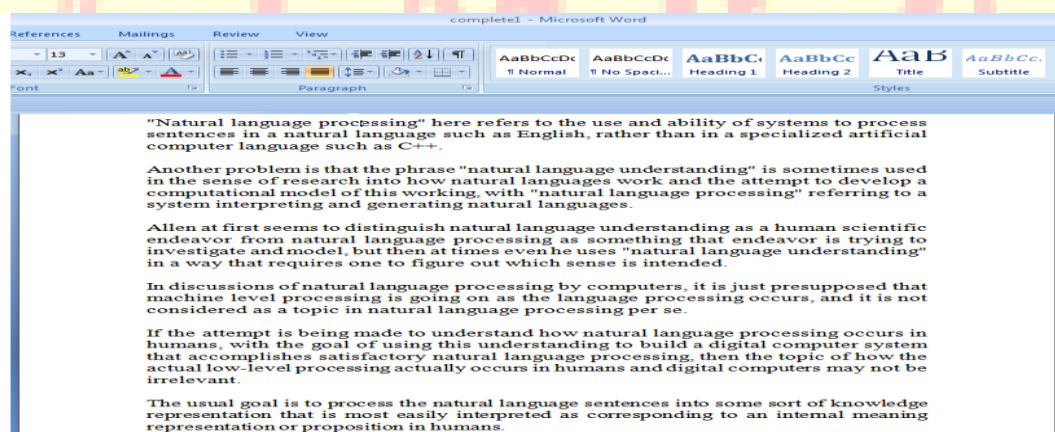
This paper proposes an optimization based model for document summarization. The model generates a summary by extracting salient sentences from a single document and reduces redundancy by measuring cosine similarity between sentences. To solve the optimization problem has been created an improved differential evolution algorithm. This algorithm can adjust crossover rate adaptively according to the fitness of individuals. We have implemented the proposed model on single document summarization task. The experimental results provide strong evidence that the proposed optimization based approach is a viable method for document summarization.

Here we have generated a summary by taking a paragraph from our system and from an automatic summarizer and then compared it. In our system, to reduce the length of the summary we are taking a threshold value (depending upon the weight of each summary sentence). If weight of the sentence is less than the threshold value, then it should be deleted from the summary. After comparing our summary with Automatic Summarizer and Microsoft word summarizer our summary gives meaningful information as well as the better summary than the summary generated by the Automatic summarizer and Microsoft word summarizer.

The summary generated by our proposed system is given below in figure 13

Summary generated according to the no of sentences entered in the summarizer is given below in figure 14

Summary generated according to the percentage of total sentence from the original document by using Microsoft word summarizer is given below in figure 15.



**Figure 13: Summary generated using DE algorithm**

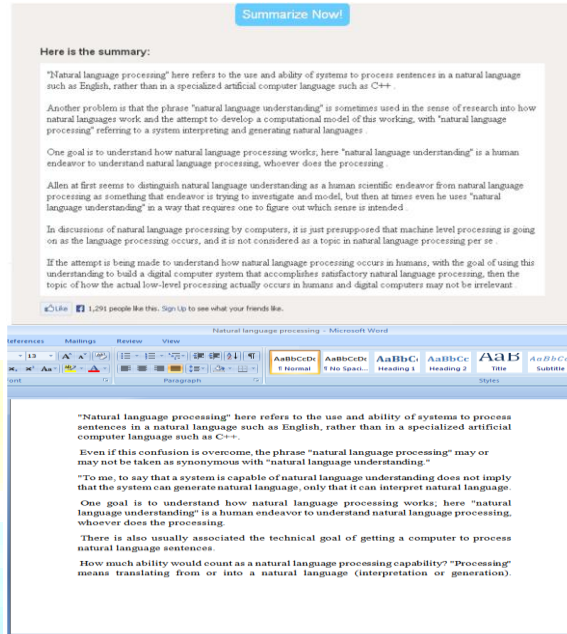


Figure 14: Summary generated by automatic summarizer Figure 15: Summary generated by MS word summarizer

## 8. Conclusion

Evaluation result of Document Summarization using DE algorithm is an effective method. DE algorithm is used to extract the sentences to form summary and it is based on a fitness function formed by two factors. The best performance of the system would be when a user exactly knows how to define  $\beta$  and  $\gamma$ , so that the result and summary would suit for his/her requirements. We take sentence similarity into account while designing the evaluation function for DE, which is helpful to improve the performance of summarization. Also there is a lot of room for improvement.

We will work on multi-document summarization and other algorithms in our future work to improve the performance of summary.

## References

- [1] Yan-Xiang He, De-Xi Liu, Dong-Hong Ji, Hua Yang, Chong Teng.(2006), MSBGA: A Multi- Document Summarization System Based on Genetic Algorithm”, International Conference on Machine Learning and Cybernetics, pp. 2659 - 2644.

- [2] Xiaojuan Zhao, Jun Tang.(2010), "Query-focused Summarization Based on Genetic Algorithm", International Conference On Measuring Technology and Mechatronics Automation, pp. 968 - 971.
- [3] Cristina Lopez-Pujalte, Vicente P. Guerrero-Bote, Cristina Lopez-Pujalte, Vicente P. Guerrero-Bote. (2003), "Order-Based Fitness Functions for Genetic Algorithms Applied to Relevance Feedback", Journal Of The American Society For Information Science And Technology, 54(2):pp. 152 - 160.
- [4] You Ouyang, Wenjie Li, Qin Lu, Renxian Zhang.(2010), "A Study on Position Information in Document Summarization", poster volume, pp. 919 - 927.
- [5] A. P. Siva kumar, Dr. P. Premchand and Dr. A. Govardhan.(2011), "Query- Based Summarizer Based on Similarity of Sentences and Word Frequency", International Journal of Data Mining and Knowledge Management Process (IJDMP), 1(3).
- [6] R.Kowsalya , R.Priya and P.Nithiya.(2011), "Multi Document Extractive Summarization Based On Word Sequences", International Journal of Computer Science, pp. 510 - 517.
- [7] Chin-Yew Lin and Eduard Hovy.(2002), "From Single to Multi-document Summarization: A Prototype System and its Evaluation", Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, pp. 457 - 464.
- [8] Rasim ALGULIEV and Ramiz ALIGULIYEV.(2009), "Evolutionary Algorithm for Extractive Text Summarization", Information Management International Conference on Machine Learning and Cybernetics, 1(2):pp. 128 - 138.
- [9] Xiaogang Ji.(2008), "Research on the Automatic Summarization Model based on Genetic Algorithm and Mathematical Regression", International Symposium on Electronic Commerce and Security, pp. 488 - 491.
- [10] Tadashi Nomoto.(2007), "Discriminative Sentence Compression with Conditional Random Fields", Information Processing and Management , 43(6):pp. 1571 - 1587.
- [11] Rasim M. Alguliev, Ramiz M. Aliguliyev.(2011), "Sentence Selection for Generic Document Summarization using an Adaptive Differential Evolution Algorithm", Swarm and Evolutionary computation,1(4):pp. 213-222.
- [12] Albaraa Abuobieda, Naomie Salim, Yogan Jaya Kumar, Ahmed Hamza Osman.(2013), "Opposition Differential Based Method for Text Summarization", Intelligent Information and Database systems, pp. 487-496.



- [13] Johanna Gei .(2011), Latent Semantic Sentence Clustering for Multi- document Summarization", PhD Thesis, University of Cambridge, ISSN. 1476-2986.
- [14] Jagadeesh Jagarlamudi .(2006), Query-based Multi-document Summarization Using Language Modelling", Master's Thesis, IIIT, Hyderabad.
- [15] Rada Mihalcea,Paul Tarau. A Language Independent Algorithm for Single and Multiple Document Summarization", [www.cse.unt.edu/~rada/papers/mihalcea.ijcnlp05.pdf](http://www.cse.unt.edu/~rada/papers/mihalcea.ijcnlp05.pdf).
- [16] Knight Kevin, Marcu Daniel,(2002), Summarization Beyond Sentence Extraction: a probabilistic Approach to Sentence Compression", *Artificial Intelligence*, 139(1):pp. 91-107.
- [17] Mitra M, Singhal Amit, Buckley Chris.(1997), Automatic Text Summarization by Sentence Extraction", *ACL/EACL-97 Workshop on Intelligent Scalable Text Summarization* , Madrid, Spain, pp. 31-36.
- [18] R. Storn, K. price.(1997), Differential evolution- a simple and efficient heuristic for global optimization over continuous spaces", *Journal of Global Optimization*, 11(4):pp. 341-359.
- [19] D. Bollegala, Y. Matsuo, M. Ishizuka.(2007), Measuring semantic similarity between words using web search engines ", *Proceedings of 16th World Wide Web Conference(WWW16)*,Alberta, Canada, pp. 757-766.
- [20] R. Mihalcea, H. Ceylan.(2007), Explorations in Automatic Book Summarization", *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing(EMNLP-CoNLL'07)* , Prague, Czech Republic, pp. 380-389.
- [21] Zhanying He, Chun Chen, Jiajun Bu, Can Wang, Lijun Zhang.(2012), Document Summarization Based on Data Reconstruction", *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence* , pp. 620-626.
- [22] R. Varadarajan, V. Hristidis.(2006), A system for query specific document summarization", *Intelligent Information and Database systems* , pp. 622-631.
- [23] K. Knight, D. Marcu .(2000), Statistics-based summarization step one: Sentence compression", *The American Association for Artificial Intelligence Conference (AAAI)* , pp. 703-710.