Sādhanā (2019) 44:110
https://doi.org/10.1007/s12046-019-1082-4

A novel approach for text summarization using optimal combination of sentence scoring methods

PRADEEPIKA VERMA* and HARI OM

Department of Computer Science and Engineering, Indian Institute of Technology (Indian School of Mines), Dhanbad 826004, India
e-mail: pradeepikav.verma093@gmail.com; hariom4india@gmail.com

MS received 13 November 2017; revised 30 May 2018; accepted 3 January 2019; published online 11 April 2019

Abstract. In this paper, a novel multi-document summarization scheme based on metaheuristic optimization is introduced that generates a summary by extracting salient and relevant sentences from a collection of documents. The proposed work generates optimal combinations of sentence scoring methods and their respective optimal weights to extract the sentences with the help of a metaheuristic approach known as teaching–learning-based optimization. In addition, the proposed scheme is compared to two summarization methods that use different metaheuristic approaches. The experimental results show the efficacy of the proposed summarization scheme.

Keywords. Multi-document summarization; word embedding; TLBO; cohesion; readability; non-redundancy.

1. Introduction

1.1 *Backgorund*

Due to the rapid growth of online information, it has become difficult to extract relevant information from a large document. Automatic text summarization (ATS) has attracted substantial interest in providing relevant information in less time [1]. Text summarization is a process of transfiguring a huge text in concise form. Traditionally, ATS is classified into extractive and abstractive summarizations. The extractive summarization [2] aims at extracting most relevant information from a given text, whereas the abstractive summarization rephrases the relevant information or generates new sentences from a group of relevant concepts in a text document. ATS can be designed for a single document or multiple documents. The former extracts the information from a single text, whereas the latter extracts the information from multiple texts written about the same topic. It is worth noting that in multi-document summarization, every topic from multiple perspective can be described by a single concise document; however, it is a more challenging task than the single document summarization in terms of the topic multiplicity and non-redundancy.

1.2 *Motivation*

This work focuses on extraction-based multi-document summarization task. The extractive summarization methods usually follow the computation of sentence salience scores using different methods to generate summary [3]. Oliveira *et al* [4] suggest that different combinations of these methods (using heuristic/simple methods) may produce better results. The heuristic approaches are often too greedy and thus they get trapped into local optimum solutions. The metaheuristic approaches are better to find the global optimum solutions. Inspired by this, we intend to investigate the optimal combinations of sentence scoring methods by modelling as an optimization problem and attempt to globally solve the problem. In recent years, many metaheuristic approaches like Particle Swarm Optimization (PSO) [5–11], Genetic Algorithm (GA) [12–15], Harmony Search Algorithm (HSA) [16], Cat Swarm Optimization (CSO) [17], Cuckoo Search (CS) [18], Multicriteria Optimization (MCO) [19] and Jaya [20] have been used to find the optimal weights for scoring methods or relevant sentences for summary generation. These metaheuristic approaches require significant computational effort for tuning a large number of controlling parameters. Further, a minor change in the algorithm parameters changes the effectiveness of the algorithm. Due to this fact, the performances of these approaches are highly inconsistent. To mitigate these problems, the notion of Teaching–Learning-based Optimization (TLBO) is used that involves a minimal number of controlling parameters and thereby provides high consistency in the performance. This motivates us to apply TLBO for accomplishing the task of text summarization.

The main challenge of multi-document summarization is to produce redundant-content-free summary generation. A

*For correspondence

summary is considered good if it represents the whole information of document [21]. Hence, the content of a summary should have minimum redundancy to maximize the coverage of information. Rautray and Balabantaray [17] have addressed this problem by employing WordNet that tells about the relations between the words [22, 23]. However, the WordNet does not provide similarity and relations between the words. The concept of word embedding can handle the task of finding the similarity and relations between the words in a better way. This motivates us to find word embedding using the word2vec tool and remove the redundancy in the summary using the word-embedding-based similarity.

1.3 *Our contribution*

In this paper, we propose a novel text summarization method that aims to find the optimal combinations of sentence scoring methods and generate summary by optimizing three important aspects of summary, which are cohesion, non-redundancy, and readability. The contributions of this paper may be summarized as follows.

a. We derive a new formula for sentence position (one of the significant sentence scoring methods) based on the heuristic that the first few and last few sentences of a document are more relevant than any other sentences.
b. The optimal combinations of the sentence scoring methods are generated using metaheuristic approach. Here, we have used TLBO metaheuristic approach. Moreover, to our best knowledge, 'TLBO' has not been applied in the field of text summarization as illustrated in table 1.
c. The performance of the proposed method is evaluated on DUC datasets using various evaluation metrics such as recall, precision, *F*-score, readability, cohesiveness and non-redundancy.
d. A comparative evaluation of the proposed scheme to metaheuristic-based text summarization methods: PSO–GA (hybrid form of PSO and GA) [5] and CSO [17] is done.

The organization of the remaining paper is as follows. Section 2 reviews the state-of-the-art methods in the field of metaheuristic-based extractive summarization and summary evaluation. Section 3 provides the related backgrounds. Section 4 introduces the proposed scheme and section 5 discusses the data and evaluation methods. Section 6 provides the results and discussion and finally the paper is concluded in section 7.

2. Literature review

The multi-document summarization has several issues like redundancy, multiple document compression, speed of sentence extraction, sentence selection, etc. In the past, some researchers have used the optimization algorithms such as GA, PSO, harmony search (HS), differential evolution (DE), CSO, etc. to generate the multi-document summaries. Possibly, the first optimization algorithm used in multi-document summarization task is GA [24], which retrieves the relevant sentences based on four good summary factors: desirable length, high coverage, high informativeness and low redundancy. It takes into account the similarity between the words using the WordNet. Kogilavani and Balasubramanie [14] present a method for optimal summary generation by grouping the related documents into the same cluster using some clustering algorithm and extracting important sentences from each cluster using the GA. Next, Khan *et al* [13] discuss a framework for abstractive summarization using the GA. In this method, the semantic role labelling has been employed to represent the source document in predicate argument structures and then the GA is used to generate the optimal weights of text features. An MCO-based multi-document summarization [19] finds the extractive generic summary with maximal relevance and minimal redundancy. The sentences are scored using five features: *tf–idf*, aggregate cross sentence similarity, title similarity, proper noun and sentence length.

Table 1. Evolutionary methods used to solve text summarization problem.

Authors	Year	Evolutionary method employed	Input document type
Gordon [12]	1988	GA	Single
He *et al* [24]	2006	GA	multi
Shareghi and Hassanabadi [16]	2008	HS	Single
Aliguliyev [25]	2009	DE	Single
Binwahlan *et al* [8]	2009	PSO	Single
Binwahlan *et al* [9]	2009	PSO	Single
Binwahlan *et al* [10]	2010	PSO	Single
Kogolavani and Balasubramanie [14]	2010	GA	Multi
Algiliev *et al* [6]	2011	PSO	Multi
He *et al* [26]	2011	Manifold ranking	Multi
Alguliev *et al* [27]	2012	DE	Multi
Mendoza *et al* [28]	2013	MA	Single
Asgari *et al* [7]	2014	PSO	Single
Khan *et al* [13]	2015	GA	Multi
Meena and Gopalani [15]	2015	GA	Single
Abbasi-ghalehtaki *et al* [5]	2016	PSO, GA	Single
Ansamma *et al* [19]	2017	MCO	Multi
Rautray and Balabantaray [18]	2017	CS	Multi
Rautray and Balabantaray [17]	2017	CSO	Multi

Rautray and Balabantaray [17] present a multi-document summarization method using a new metaheuristic approach, called 'cat swarm optimization'. In this method, the multiple documents are reduced into a single document via sentence informative score and inter-similarity score. The sentence informative score is calculated using the *tf–idf* vectors and the inter-similarity score is calculated using the cosine similarity. Rautray and Balabantaray [18] discuss a technique for multi-document summarization using the CS employing three features: coverage, cohesion and readability. Alguliev *et al* [27] discusses a multi-document summarization technique by considering three aspects of summarization: content coverage, diversity, and length of summary; and optimizes these aspects using the differential evolution. Alguliev *et al* [6] introduce a PSO-based text summarization model that maximizes the content coverage and minimizes the redundancy in the summary. It optimizes relevancy, redundancy and summary length, together using the binary PSO.

The multi-document summarization also has the issue of finding similarity between the documents. Some researchers have discussed the similarity measurement using the word embedding. The method [29] introduces a Word Mover Distance (WMD) function between the text documents in which the embedded words of one document need travel to reach the embedded words of other document. The scheme [30] discusses a similarity function on the basis of embeddings in the form of a negative dissimilarity function using the dissimilarity function. The scheme [31] presents an embedding-based similarity function for short texts. This discussion shows that the metaheuristic approaches and word-embedding-based similarity measurements have been widely used in text summarization. Motivated by this discussion, we present a novel TLBO-based multi-document summarization method with word embedding.

3. Preliminaries

In this section, we provide the related preliminaries used in this paper.

3.1 *Word2vec tool*

Word2vec [32] is a set of related models that provide an efficient implementation of *continuous bag-of-words* (*CBOW*) and *skip-gram* models. These models are used to produce the word embeddings in the form of vector spaces with several hundreds of dimensions of features, where each word of a document is represented as one-hot vector with dimension $1 \times nd$, where nd is vocabulary size. Our scheme focuses on the skip-gram neural network model that works on a single hidden layer. Further, the skip-gram model trains each word by looking up at nearby words in a document and randomly picks up one. Here, we use the pre-trained word2vec model on Google news dataset that contains about 100 billion words and 3000000 unique phrases built with layer size of 300. The effectiveness of word embedding is limited by Out-Of-Vocabulary (OOV) words. Hence, we handle these words by a mimicking approach using sub-word RNNs [33]. It does not require re-training on the original word embedding corpus; instead, learning is performed at type level. For example the embedding of word 'exaggerate' will be obtained by pairing each character of the word (like 'ex', 'exa', 'rate', 'ge', etc.) and then finding the related words for each pair using a model that has been trained by pairing the characters of each word for which embeddings existed. In this way, the related words for 'exa' can be 'over', 'much', etc.; for 'rate' can be 'measure', 'estimate', 'value', etc. Therefore, the related words for 'exaggerate' can be 'overestimate', 'overvalue', etc.

3.2 *TLBO*

TLBO is a new metaheuristic approach [34]. Its concept stems from the impact of a teacher on the students and conversation between students. A teacher is considered as a person having the best knowledge who can teach the students and make them equivalently knowledgeable. This effective feature is the origin of TLBO. This algorithm is composed of two phases: teacher phase and student phase. The student solution enhances their knowledge from the teacher solution in teacher phase and through exchanging their knowledge between themselves in student phase. In this study, TLBO is considered to address the problem of text summarization as well as finding optimal contributions of scoring methods. In teacher phase, TLBO explores the search space of a given document to find the most relevant sentence of a document and subsequently optimal combination of sentence scoring methods. In student phase, TLBO finds relevant sentences of the document according to the best sentence found in teacher phase. In this way, these phases of TLBO are correlated with text summarization.

3.3 *Similarity measure*

Generally, the sentences of one document, use one word effectively to say one concept. As a result, it is not necessary to analyse the relations between two words belonging to different sentences of one document. Hence, we measure the similarity between the sentences of one document using the *tf–idf*-based cosine similarity function by counting the words or sequence of words. On the other hand, the sentences of two different documents may use different words to say one concept. As a result, it is necessary to analyse the relations between two words belonging to different documents. Hence, in this case, we have been using a recently introduced similarity function based on WMD [29].

3.3a *Cosine similarity:* It uses the word's score representation of textual units. The score of each word is calculated by *tf–idf*, which is a combination of word frequency and inverse sentence frequency schemes, which is calculated as follows. Assume that w_p is a word p of sentence s_j. Then

$$\alpha_{pj} = tf_{pj} \times \log \frac{|d_i|}{|d_i|_p} \tag{1}$$

where α_{pj} denotes *tf–idf* of a word w_p of sentence s_j, and $|d_i|_p$ is the number of sentences of document d_i in which word w_p occurs. According to this representation, each sentence s_j is represented as $\hat{s_j} = \{\alpha_{1j}, \alpha_{2j}, \ldots, \alpha_{|s_j|j}\}$. Then, the cosine similarity between two vectors $\hat{s_j}$ and $\hat{s_{j'}}$ is calculated as follows:

$$Sim_{Cos}(\hat{s_j}, \hat{s_{j'}}) = \frac{\sum_{p \in j,j'} \alpha_{pj}\alpha_{pj'}}{\sqrt{\sum_{p \in j} \alpha_{pj}^2}\sqrt{\sum_{p \in j'} \alpha_{pj'}^2}}. \tag{2}$$

3.3b *WMD-based similarity:* It is simply the negative distance calculated by WMD. Assume that w_p and $w_{p'}$ are two words of different documents. Their similarity is calculated as follows:

$$Sim_{WMD}(w_p, w_{p'}) = -WMD(w_p, w_{p'}). \tag{3}$$

WMD uses the word embeddings and widely studied Earth Mover's Distance [35] to find the dissimilarity between the words. It calculates the minimum amount of distance that the embedded words of one document needs to travel to reach the embedded words of another document.

4. Proposed multi-document summarization

Apart from relevant information, a good summary should contain three important features that have been considered in past years [16, 36–38]. They are the following.

1. **Readability:** The summary should be readable. A readable summary consists of sentences that elaborate their preceding sentences and contents to make it easily understandable. The summary should not contain complex words and grammatical errors.
2. **Cohesion:** The summary sentences should be conceptually related to each other.
3. **Non-redundancy:** It indicates the novelty in summary.

We would like to generate a summary such that these features are maximized. We now introduce our proposed scheme.

Let $D = \{d_1, d_2, \ldots, d_{|D|}\}$ be a given collection of documents, document $d_i = \{s_1, s_2, \ldots, s_{|d_i|}\}, i = 1, 2, \ldots, |D|$, is a collection of sentences. Each document is first preprocessed in three stages: sentence separation, stop-word removal and stemming. As a result, sentence $s_j = \{w_1, w_2, \ldots, w_{|s_j|}\}, j = 1, 2, \ldots, |d_i|$, is converted into a set of words. Each sentence s_j of document d_i is scored using sentence scoring methods $T = \{t_1, t_2, \ldots, t_n\}$, where n is number of scoring methods. As a result, each sentence s_j is transformed into vector form $\hat{s_j} = (St_1, St_2, \ldots, St_n)$, where St_k denotes score of the sentence using t_k^{th} scoring method, $k = \{1, 2, \ldots, n\}$.

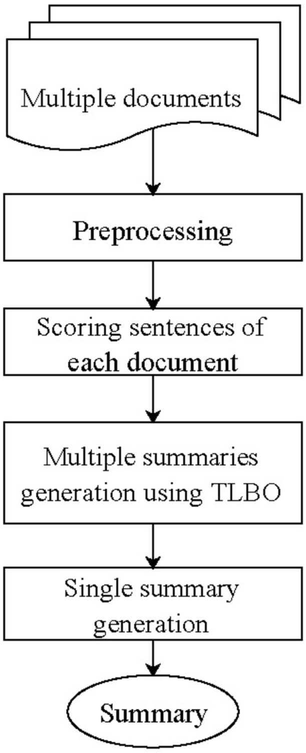

Multiple documents
Preprocessing
Scoring sentences of each document
Multiple summaries generation using TLBO
Single summary generation
Summary

Figure 1. Methodology of proposed scheme.

We attempt to find a subset of sentences of each document $d_i = \{s_1, s_2, \ldots, s_l\}$, where l is length of summary that covers the main content of the document. We accomplish this task by employing TLBO. As a result, we find a collection of summaries $Sum_s = \{Sum_{s_1}, Sum_{s_2}, \ldots, Sum_{s_{|D|}}\}$, $Sum_i \in d_i$. Finally, we find a summary $Sum_s = \{s_1, s_2, \ldots, s_L\}$, where $L \leq |D| * l$ is length of final summary obtained by minimizing the redundant content between two summaries. The methodology of our proposed approach as shown in figure 1 is discussed here.

4.1 *Preprocessing*

Preprocessing is a preliminary process of our proposed algorithm that covers the stop-word removal, stemming [39] and sentence separation steps. We accomplish these processes by employing Natural Language Tool Kit (NLTK) [40]. Here, we use the Porter stemmer for English text.

4.2 *Sentence scoring*

In order to summarize a document using metaheuristic approach, it is necessary to convert the text data into a suitable numerical form. Scoring sentences of the document using text features is the unique way to transform the sentences of the document in a vector form. The text features help in recognizing the relevant sentences in the document. Hence, they play a major role in processing of a text document. Oliviera *et al* [4] suggest some scoring methods using text features. Some of them are modified in our scheme as briefly presented here.

4.2a *Aggregate similarity* (t_1): The sentences can be scored by finding the common information residing in them. This method considers the sentences having more common information as more relevant sentences. It can be expressed as follows:

$$St_1(s_j) = \frac{\sum_{j'=1, j\neq j'}^{|d_i|} (Sim_{Cos}(s_j, s_{j'}))}{|s_j|}. \tag{4}$$

4.2b *Bushy path* (t_2): Connected directed edges indicate the number of sentences connected with a sentence. It computes the similarity of a sentence with other sentences and counts the number of sentences that have high similarity. The sentences having more outgoing connections are considered as more relevant. It can be expressed for a sentence s_j as follows:

$$St_2(s_j) = \frac{\text{number of outgoing connected edges with}\, s_j}{|d_i|}. \tag{5}$$

4.2c *Cue phrases* (t_3): The sentence with the phrases 'In the abstract', 'In conclusion' , etc. contains significant information. Hence, the scores of sentences using this method are calculated as follows:

$$St_3(s_j) = \frac{\text{number of cue phrases in}\, s_j}{|s_j|}. \tag{6}$$

4.2d *Lexical relation* (t_4): A good sentence is constructed with a set of highly related words. On the basis of this supposition, it finds a strong lexical relation (for example, 'big bang theory' is a strong chain of words) between the words of sentences using the embedding tool *word2vec*. In the following, the sentence having the highest number of strong chains of related words gets the highest score and it is calculated as

$$St_4(s_j) = \frac{\text{number of strong chain in}\, s_j}{|s_j|}. \tag{7}$$

4.2e *Named entities* (t_5): The method considers that the sentences containing named entities have higher probability to be included in the summary and it is calculated as

$$St_5(s_j) = \frac{\text{number of named entities in}\, s_j}{|s_j|}. \tag{8}$$

4.2f *Noun and verb phrases* (t_6): The noun and verbal phrases generally refer to the subject and predicate parts of a sentence. The noun phrase includes group of nouns and their components. The verb phrase includes helping verbs, their components and objects. This method considers that a sentence's significance is determined by its number of nouns and verb phrases. It can be computed as follows:

$$St_6(s_j) = \frac{\text{number of noun and verb phrases in}\, s_j}{|s_j|}. \tag{9}$$

4.2g *Numerical data* (t_7): The sentences with numerical data generally indicate important information, which is likely to be included in the summary. It is calculated as follows:

$$St_7(s_j) = \frac{\text{number of numerical data in}\, s_j}{|s_j|}. \tag{10}$$

4.2h *Open relations* (t_8): This method considers that a sentence containing open relations like quotation marks, parenthesis, etc. may indicate a sentence with more significant facts. It can be computed as follows:

$$St_8(s_j) = \frac{\text{number of open relations in sentence}\, s_j}{|s_j|}. \tag{11}$$

4.2i *Proper noun* (t_9): A proper noun indicates an individual person, place or organization, and it conveys better information in a sentence than any other feature. It is calculated as follows:

$$St_9(s_j) = \frac{\text{number of prope r nouns in sentence}\, s_j}{|s_j|}. \tag{12}$$

4.2j *Sentence centrality* (t_{10}): The sentence centrality refers to the degree of word overlap of a sentence with other sentences. This method considers that the central sentences best describe the significant information of a document. It can be computed as follows:

$$St_{10}(s_j) = \frac{\sum_j w_{pj} \cap w_{pj'}}{\sum_{j'} w_{pj} \cup w_{pj'}}. \tag{13}$$

4.2k *Sentence length* (t_{11}): This feature is used to filter out very short and very long sentences. The average length of a sentence can be calculated by the following function:

$$AL(d_i) = \frac{\min(|d_i|) + \max(|d_i|)}{2} \tag{14}$$

where $AL(d_i)$ is the average length of a sentence in the summary. On the basis of average length, we calculate the sentence length score as follows:

$$St_{11}(s_j) = 1 - \frac{|AL(d_i) - |s_j||}{\max(|d_i|)}. \tag{15}$$

4.2l *Sentence position* (t_{12}): It is one of the major features for sentence extraction. Generally, in a document, first few sentences and last few sentences consist of more significant information than others. Hence, we take into account this factor and calculate the score by deriving a new formula as follows:

$$St_{12}(s_j) = \frac{|(|d_i|/2) - j|}{(|d_i|/2)}. \tag{16}$$

4.2m *Sentence with title words* (t_{13}): The title words in sentences show highly relevant sentences of a document. This score is calculated as

$$St_{13}(s_j) = \frac{\left(\sum_{w_{p'}\in title}\sum_{w_p\in s_j} w_{p'}\cap w_p\right)^2}{|title| \times |s_j|} \tag{17}$$

where |*title*| is number of words in the title.

4.2n *Sentence significance* (t_{14}): Sentence significance tells about the information contribution. The significance score for a sentence is calculated as follows:

$$St_{14}(s_j) = \frac{\sum_{p=1}^{|s_j|}\alpha_{pj}}{|s_j|}. \tag{18}$$

4.2o *Upper case word* (t_{15}): The sentences with upper case words usually have significant information. These words may refer to acronyms, organization names, etc. The sentences are scored based on this method as follows:

$$St_{15}(s_j) = \frac{\text{number of upper case keywords in } s_j}{|s_j|}. \tag{19}$$

4.2p *Frequent words* (t_{16}): A sentence with frequent words can also represent the title of the document and it conveys significant information. It is calculated by finding the term frequency (*tf*) of words. Top 30% highest frequency words are considered as frequent words. The score of a sentence with frequent words is calculated as follows:

$$St_{16}(s_j) = \frac{\sum_{w_{p'}\in fw}\sum_{w_p\in s_j} w_{p'}\cap w_p}{|s_j|} \tag{20}$$

where *fw* denotes frequent words.

4.3 *Multiple summaries generation using TLBO*

The summaries with respect to each document of a multi-document are generated with two aspects. (i) First is optimal combinations of sentence scoring methods. Oliveira *et al* [4] suggest that the set of combinations of sentence scoring methods can produce better results. In our study, every combination is a set of four sentence scoring methods. ii) Second is an optimal weight for each sentence scoring method, which also produces good results. On the basis of these two aspects, the summaries are generated for each document of multi-document using TLBO as illustrated in figure 2. The steps involved are described here.

4.3a *Initialization:* The initial weights of text features are generated randomly in the form of a solution $X = \{x_1, x_2, \ldots, x_y\}$ with size y and dimension *ND* in the range (0, 1). The termination criterion (z) is initialized as illustrated in figure 3. Here, we consider 16 text features. Hence, *ND* of every solution is 16.

4.3b *Mean of scoring methods' weights:* The adopted metaheuristic approach relies on the fact that a good solution produces a better mean for the results of a learner. Hence, each solution is updated using the mean of the generated weights $M = \{m_1, m_2, \ldots, m_{ND}\}$ with respect to each scoring method as illustrated in figure 4.

4.3c *Sentence extraction:* After calculating the mean of each sentence scoring method, we sort the solutions according to their respective initial weights. Hence, each solution, say x_q, is divided into four sub-solutions to obtain different combinations of four sentence scoring methods. In this way, we find four sets of sub-solutions, say, $x_q = \{x_{q1}, x_{q2}, x_{q3}, x_{q4}\}$ for each solution. Next, each sub-solution with its respective weight is used to extract a set of sentences as a summary from the document. As a result, we get four summaries for each solution, say $Sum_{s_q} = \{Sum_{s_{q1}}, Sum_{s_{q2}}, Sum_{s_{q3}}, Sum_{s_{q4}}\}$, where Sum_{s_q} represents the system summaries generated using x_q solution. These four summaries are then evaluated using a fitness function. The summary that has the maximum fitness score$F(Sum_{s_q}) = \max\{F(Sum_{s_{q1}}), F(Sum_{s_{q2}}), F(Sum_{s_{q3}}), F(Sum_{s_{q4}})\}$is considered as the best summary among four for the solution and its fitness score is assigned as the local best $lb_q = F(Sum_{s_q})$ for the solution. The sub-solution corresponding to the best summary is considered as the best optimal combination among four.

4.3d *Fitness function:* A good summary can be generated by considering three major aspects: cohesion, non-redundancy and readability. Cohesion indicates a conceptual relation between the sentences of a summary. It means that every sentence of a summary should discuss about the same information. A good summary does not contain redundant content. It causes decreasing coverage of significant information in summary. Readability indicates linking of sentences with their preceding sentences. A good summary should be readable, so that flow of information could be maintained. Shareghi and Hassanabadi [16] suggest cohesion estimation as follows:

$$Cohesion(Sum_s) = \frac{\log(Avg_{s_j\in\{Sum_s\}}(Sim_{Cos}(s_j)) \times 9 + 1)}{\log(\max_{s_j\in Sum_s}(Sim_{Cos}(s_j)) \times 9 + 1)} \tag{21}$$

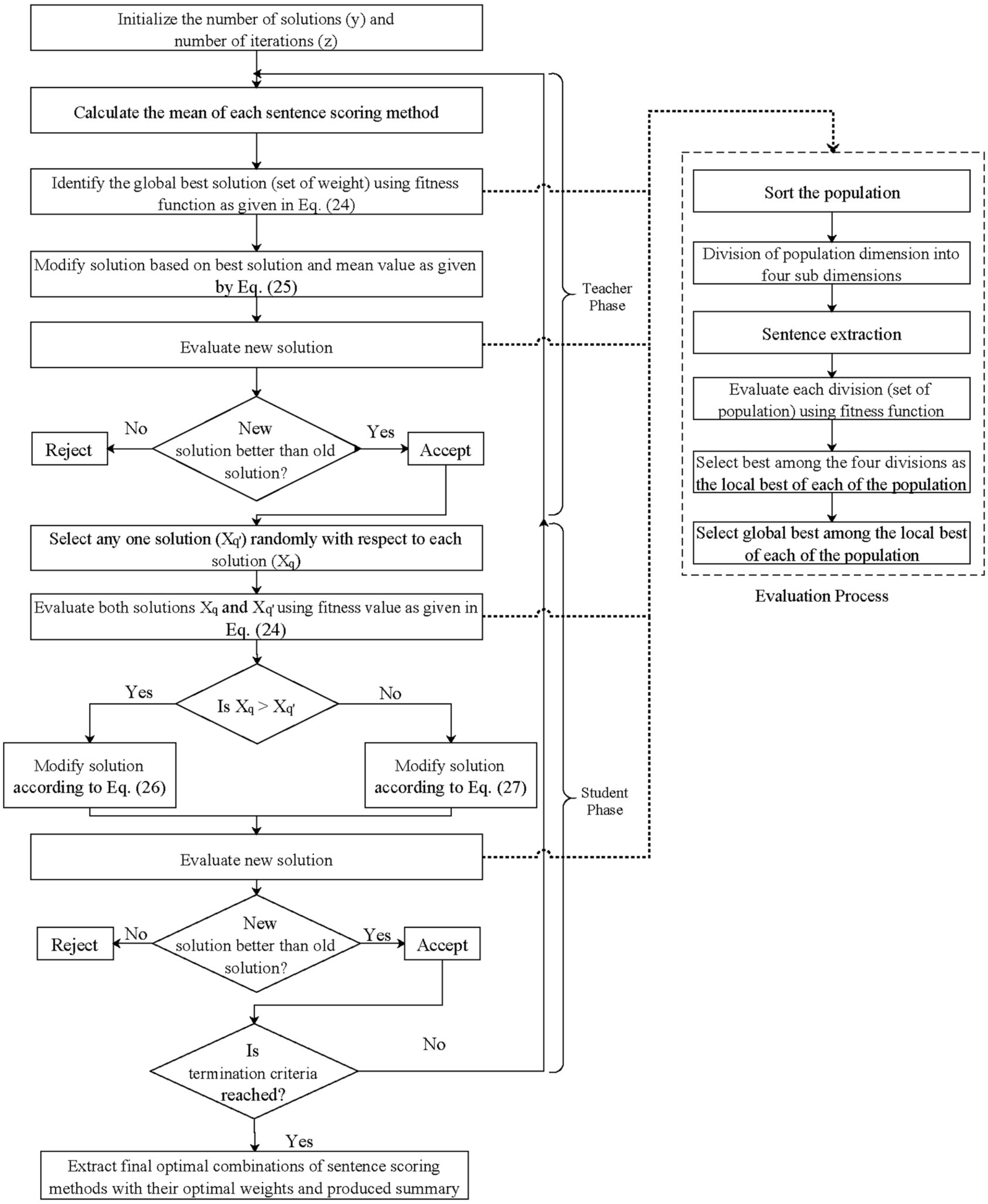

Initialize the number of solutions (y) and number of iterations (z)
Calculate the mean of each sentence scoring method
Identify the global best solution (set of weight) using fitness function as given in Eq. (24)
Modify solution based on best solution and mean value as given by Eq. (25)
Evaluate new solution
Reject
No
New solution better than old solution?
Yes
Accept
Teacher Phase
Select any one solution (Xq') randomly with respect to each solution (Xq)
Evaluate both solutions Xq and Xq' using fitness value as given in Eq. (24)
Yes
Is Xq > Xq'
No
Modify solution according to Eq. (26)
Modify solution according to Eq. (27)
Student Phase
Evaluate new solution
Reject
No
New solution better than old solution?
Yes
Accept
Is termination criteria reached?
No
Yes
Extract final optimal combinations of sentence scoring methods with their optimal weights and produced summary
Sort the population
Division of population dimension into four sub dimensions
Sentence extraction
Evaluate each division (set of population) using fitness function
Select best among the four divisions as the local best of each of the population
Select global best among the local best of each of the population
Evaluation Process

Figure 2. Flow chart for multiple summaries generation using TLBO.

where similarity has been calculated using cosine similarity. $Avg_{s_j \in \{Sum_s\}}$ is the average of the similarities of all sentences belonging to system summary and $\max_{s_j \in Sum_s}(Sim(s_j)$ is the maximum of similarities in system summary.

Non-redundancy [19] can be obtained by finding dissimilarity between the sentences using cosine similarity as follows:

$$Non\_redundancy(Sum_s) = 1 - \max_{j \in Sum_s}(Sim_{Cos}(s_j, s_{j'})) \tag{22}$$

where Sum_s denotes system summary and $Sim(s_j, s_{j'})$ denotes similarity between s_j and $s_{j'}$.

Readability [16] can be calculated by finding the similarity of a sentence with its preceding sentence using the cosine similarity:

$$Readability(Sum_s) = \frac{\sum_{j \in Sum_s} Sim_{Cos}(s_j, s_{j+1})}{\max Sim_{Cos}(s_j)}. \tag{23}$$

Finally, the fitness function is formulated by the linear combination of these aspects with their weights as follows:

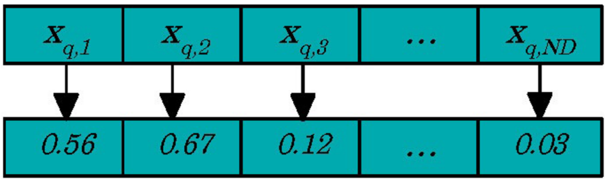


Figure 3. Initialization of solution.

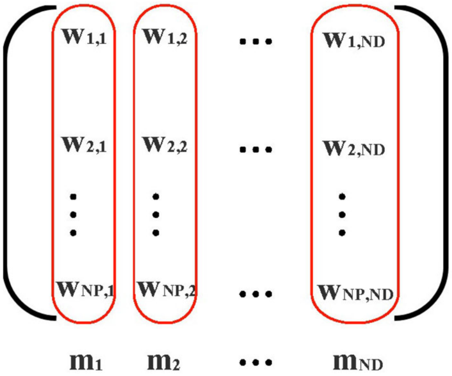


Figure 4. Mean of scoring methods' weights.

$$F(Sum_s) = \beta \times Cohesion(Sum_s) + \left(\frac{1-\beta}{2}\right) \times Non\_redundancy(Sum_s) + \left(\frac{1-\beta}{2}\right) \times Readability(Sum_s). \tag{24}$$

Here, the value of $0<\beta<1$ is user defined. We assume that the sentences of a summary should have high conceptual relation to each other. Hence, we give higher weight to cohesion than others. For the rest of the two, we give equal weights. We use the value of β as 0.4 and the weights for others are 0.3 and 0.3.

4.3e *Identification of teacher:* A teacher is considered as the most knowledgeable person who can produce the best result and enhance the knowledge of others. With this consideration, a solution that produces the best result $Best\_X = \{lb_1, lb_2, \ldots, lb_y\}$ is assigned as a teacher. The teacher will try to shift the mean M toward $Best\_X$. Hence, $Best\_X$ will act as New_M. The evaluation of each solution is done using a suitable fitness function as described earlier.

4.3f *Update weights of text features in teacher phase:* The weights of text features are updated with the help of assigned teacher solution and the mean values of text features as follows:

$$New\_X_q = X_q + r(Best\_X - T_F \times M) \tag{25}$$

where New_X denotes the updated q^{th} solution of weights. X_q denotes the existing q^{th} solution of the weights; r is a random number in the range [0,1] and T_F is the teaching factor, which is either 1 or 2. M is the existing mean. The updated weights are then evaluated using the fitness function. If a solution produces better result than the previous one, then the updated weights are accepted. This modification in weights is done in teacher phase.

4.3g *Update weights in learner phase:* The solution can also be enhanced by gaining the information from the solution that produces better result that is known in learner phase. Hence, the weights of text features are updated in this phase by randomly selecting one for each solution. Let, for each x_q, $x_{q'}$ be the randomly selected solution. Then, by evaluating both the solutions x_q and $x_{q'}$, if $x_q < x_{q'}$, x_q is updated as follows:

$$New\_x_q = x_q + r_z(x_q - x_{q'}). \tag{26}$$

Otherwise, x_q is updated as follows:

$$New\_x_q = x_q + r_z(x_{q'} - x_q) \tag{27}$$

where New_x_q represents new q^{th} solution.

If $x_q = x_{q'}$, then x_q remains the same. In this way, if any modification in weights of the text features occurs, then it is evaluated. If it is found to be a better solution than the previous one, then the solution is updated; otherwise, it remains unchanged.

4.3h *Termination of the algorithm:* If the number of user-defined iterations is over, then the best weights of text features that have the highest fitness value are selected as final weights of the scoring methods. If the number of iterations is not over, then the new weights are calculated to update the weights using Eqs. (25)–(27).

4.4 *Single summary generation*

Single summary from multiple summaries of multiple documents can be generated by merging all of them. Since multi-document summarization has a major challenge of appearance of redundant content in a summary due to redundancy in multiple documents, the generated summaries are analysed by semantic similarity measure between the inter-summary sentences to reduce the redundancy between summaries as follows.

The overall similarity of sentence s_j of summary Sum_{s_i} with a sentence $s_{j'}$ of another summary $Sum_{s_{i'}}$ is calculated as follows:

$$Sim(s_j) = \sum_{j'=1}^{l'} Sim_{WMD}(s_j, s_{j'}),$$

$$Sim_{WMD}(s_j, s_{j'}) = \frac{\sum_{w_p \in s_j} \sum_{w_{p'} \in s_{j'}} Sim_{WMD}(w_p, w_{p'})}{|s_j|.|s_{j'}|},$$

$$s_j \in Sum_i, s_{j'} \in Sum_{i'}, 1 \le i, i' \le |D|, \text{and } i \ne i', \tag{28}$$

where l' is the length of summary $Sum_{s_{l'}}$. For normalizing the similarity score, we use the max–min normalization, as follows:

$$Sim(s_j)_{norm} = \frac{Sim(s_j) - \min\{Sim(s_j)\}}{\max\{Sim(s_j)\} - \min\{Sim(s_j)\}}. \tag{29}$$

Usually, a similarity function calculates *tf–idf* to find the similarity between the sentences. However, this feature is not often suitable for inter-summary sentences similarity due to frequent near-orthogonality [41]. It also does not capture the similarity between the individual words. The summaries of two different documents may extract two sentences that have no common words, but may be related to each other. Hence, it is necessary to find the similarity of each word in sentences. To get rid of this problem, we apply the WMD-based similarity, whose performance is based on the word embeddings.

Finally, the sentences with least similarity from all documents are extracted using a cut-off value. The cut-off value is decided through an experimental analysis, where we have done text summarization using some baseline methods at different summary lengths like 10%, 20%, 30%, 40% and 50%. We have found that at 30% summary length, these methods give the best performance. Hence, we have extracted 30% sentences from each document to generate a summary of 30% length. The sentences having similarity less than the given cut-off value are considered for inclusion in a document. All non-redundant sentences from all summaries are merged into one to generate a single summary.

5. Experimental set-up and dataset

This section discusses the experimental set-up for the proposed scheme, dataset used for evaluation, evaluation metrics and performance analysis.

5.1 *Experimental set-up*

Experiments are done using the ROUGE tool [42], which evaluates a summarization system on the basis of *N*-gram, skip-bigram plus unigram, skip-bigram, longest common sequence and weighted longest common sequence. It evaluates the performance in terms of three metrics: precision, recall and *F*-score. We also evaluate the readability and cohesion factor of the system summary.

5.2 *Dataset*

The DUC 2006 and DUC 2007 datasets are used for evaluation, which contain 50 and 45 sets of documents, respectively, and each set contains 25 documents. A brief discussion about these datasets is given in table 2.

5.3 *Evaluation metric*

The summaries are evaluated using the co-selection-based metrics and content-based metrics. The precision, recall, F_1 and accuracy metrics are adopted for co-selection-based evaluation, which is accomplished with the reference summary. Suppose D is a document, RC denotes its retrieved correct sentences, NI denotes the non-retrieved incorrect sentences, NC denotes the non-retrieved correct sentences and RI denotes the retrieved incorrect sentences; then the *recall*, *precision*, F_1 and *accuracy* are defined as follows.

1. Recall: Recall is the ratio of total retrieved correct sentences to the total of retrieved and non-retrieved correct sentences in a document. Mathematically, it is given as follows:

$$recall = \frac{|RC|}{|RC| + |NC|}. \tag{30}$$

2. Precision: Precision is the ratio of total retrieved correct sentences to the total of retrieved correct sentences and retrieved incorrect sentences in a document. Mathematically, it is given as follows:

$$precision = \frac{|RC|}{|RC| + |RI|}. \tag{31}$$

3. F_1 score: The F_1 score is the harmonic mean of the precision and recall. Mathematically, it is given as follows:

$$F_1 = \frac{2 \times precision \times recall}{precision + recall}. \tag{32}$$

4. Accuracy: Accuracy is the ratio of the total of the retrieved correct and non-retrieved incorrect text to the total text of a document. Mathematically, it is given as follows:

$$accuracy = \frac{|RC| + |NI|}{|D|}. \tag{33}$$

Table 2. Dataset description.

	Datasets	
Dataset parameters	DUC 2006	DUC 2007
Number of sets of documents	50	45
Number of documents in each set	25	25
Average number of sentences per document	30.12	37.5
Maximum number of sentences per document	79	125
Minimum number of sentences per document	5	9
Data source	AQUAINT	AQUAINT
Summary length (%)	40	40

The cohesion and readability metrics are adopted for content-based evaluations. The cohesion scores are evaluated using Eq. (21) and the readability of summaries is evaluated in terms of five readability metrics [43]: Flesch–Kincaid Grade Level (FKGL), Gunning Fog Index (GFI), SMOG Index (SMOGI), Coleman–Liau Index (CLI) and Automated Readability Index (ARI). These metrics tell how much the contents of a summary are easily understandable at word level. They are computed as follows:

$$FKGL = 0.39\left(\frac{W}{|Sum_s|}\right) + 11.8\left(\frac{Syl}{W}\right) - 15.59 \qquad (34)$$

$$GFI = 0.4\left[\left(\frac{W}{|Sum_s|}\right) + 100\left(\frac{cw}{W}\right)\right] \qquad (35)$$

$$SMOGI = 1.0430 \times \sqrt{\left(Syl \times \frac{30}{|Sum_s|}\right)} + 3.1291 \qquad (36)$$

$$CLI = 0.0588 \times AC - 0.296 \times AS - 15.8 \qquad (37)$$

$$ARI = 4.71\left(\frac{C}{W}\right) + 0.5\left(\frac{W}{|Sum_s|}\right) - 21.43 \qquad (38)$$

where W denotes the total words; Syl and cw are total syllables and complex words (those words that are not used frequently in normal writing), respectively. AC is the average number of characters C per 100 words, and AS is the average number of sentences per 100 words. Additionally, the sentence level readability RPS (Relatedness with Previous Sentence) is analysed using Eq. (23) and the results of readability metrics are shown in table 5.

6. Results and discussion

The results of optimal features' sets using the TLBO have been evaluated as shown in tables 3 and 4. These tables contain different combinations of the scoring methods used to extract the sentences for summary generation with a number of documents, whose sentences are extracted with the respective combination of the scoring methods, average recall values of a number of documents for every combination of scoring methods and the average recall values of a number of documents with all sentence scoring methods. As evident from these results, the optimal combinations of scoring methods produce better results than those that take into account all features. It is also evident that the sentence scoring methods t_2, t_4, t_{10}, t_{12} and t_{13} contribute more than others in sentence extraction.The proposed model has been evaluated in the context of multi-document extractive summarization task on the datasets provided by the Document Understanding Conference. For each data collection, the proposed scheme generates a summary at 30% summary length.

Table 3. Document's summary description and their evaluation with respect to the set of sentence scoring methods on DUC 2006 dataset.

Set of features	Features description	Number of documents	Recall values of summaries extracted from respective features	Recall values of summaries extracted from all features
Set1	t_2, t_{12}, t_{14}, t_{16}	261	0.646	0.610
Set2	t_2, t_4, t_{10}, t_{12}	192	0.615	0.597
Set3	t_1, t_5, t_{13}, t_{14}	166	0.672	0.624
Set4	t_4, t_5, t_{12}, t_{13}	138	0.658	0.603
Set5	t_1, t_2, t_{10}, t_{13}	123	0.579	0.561
Set6	t_5, t_{10}, t_{12}, t_{13}	94	0.604	0.581
Set7	t_4, t_{10}, t_{12}, t_{16}	67	0.682	0.646
Set8	Mixed	209	0.613	0.565

Table 4. Document's summary description and their evaluation with respect to the set of sentence scoring methods on DUC 2007 dataset.

Set of features	Features description	Number of documents	Recall values of summaries extracted from respective features	Recall values of summaries extracted from all features
Set1	t_2, t_{12}, t_{14}, t_{16}	278	0.682	0.635
Set2	t_1, t_4, t_{10}, t_{12}	204	0.675	0.576
Set3	t_1, t_{11}, t_{13}, t_{14}	189	0.588	0.551
Set4	t_4, t_5, t_{12}, t_{13}	143	0.612	0.595
Set5	t_1, t_2, t_{10}, t_{12}	96	0.541	0.504
Set6	t_5, t_{10}, t_{12}, t_{13}	54	0.539	0.522
Set7	Mixed	161	0.616	0.607

To compare the performance of our proposed model, we have considered other existing metaheuristic approaches: hybrid PSO–GA and CSO, with the same settings for implementation and evaluation. Table 5 shows the co-selection-based evaluation and table 6 shows the content-based evaluation.

Table 5. Co-selection-based evaluation results for summarization schemes.

Datasets	Methods	Co-selection-based performance parameters Precision	Recall	F_1	Accuracy
DUC 2006	TLBO	0.561	**0.634**	**0.595**	**0.850**
	PSO–GA	0.532	0.571	0.551	0.814
	CSO	**0.568**	0.520	0.543	0.805
DUC 2007	TLBO	**0.557**	**0.608**	**0.581**	**0.832**
	PSO–GA	0.524	0.587	0.554	0.817
	CSO	0.538	0.569	0.553	0.803

Bold values indicate the overall best performance.

The performance analysis of the summarization scheme has been done in two stages: co-selection- and content-based analysis, which is presented here.

6.1 *Co-selection-based analysis*

We have evaluated the scores by matching the unigrams between the system summary and reference summary using the ROUGE (Recall-Oriented Understudy for Gisting Evaluation) tool, as shown in table 2. The precision score for the proposed scheme is 0.561, which is the ratio of common unigrams in the system summary and reference summary to the total unigrams in the system summary. This implies that ≈56% contents of the system summary have been accurately extracted by the proposed scheme. The recall score is 0.634, which is the ratio of common unigrams in the system summary and reference summary to the total unigrams in the reference summary. This implies that ≈63% contents of the reference summary have been accurately extracted. The obtained F_1 score is 0.595, which is used to know how much contents are accurately extracted with respect to both the system summary and reference summary by giving them equal weights. This implies that ≈60% contents have been accurately extracted. Finally, the accuracy score, which takes into account the common unigrams as well as the non-retrieved non-relevant unigrams, refers to the closeness of accuracy of the extracted contents with respect to whole document. The overall accuracy is ≈85%, which shows that the proposed scheme performs effectively. Figure 5 shows the graphical representation of the co-selection-based results.

Since the fulcrum of the proposed scheme is to find the optimal set of scoring methods and their weight assignment using TLBO, to show its effectiveness, we have implemented the existing schemes for text summarization that are based on metaheuristic approaches. Abbasighalehtaki *et al* [5] and Rautray and Balabantaray [17] have discussed the summarization schemes based on the following metaheuristic approaches: hybrid PSO–GA and CSO, respectively. From table 2, we can observe that TLBO performs the best among all the schemes for the following reasons.

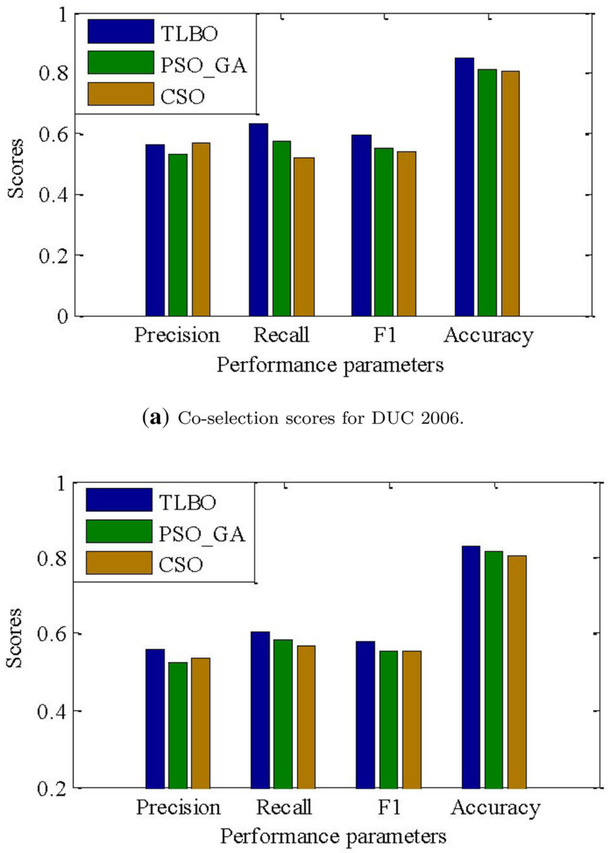
i. GA does not search the solution in local space. Hence, when the distance between the particles becomes shorter, it is unable to find the optimal solution. On the other hand, PSO searches the solution in both local and global spaces; still its performance is lower than that of TLBO. The main reason is that PSO mainly focuses on the global search space. The best solution is considered as its local best solution for every particle in PSO, whereas, in TLBO, the best solution is selected from a set of solutions obtained from the influence of a teacher solutions and learner solutions.

ii. The main advantage of TLBO is the controlling-parameter-free scheme. Generally, the evolutionary optimization approaches like GA, PSO, CSO consist of a large number of variables and constraints that cause frequent variations in their output. These approaches require tuning of the controlling parameters. TLBO requires only one random number, due to which the variation in output is very less in comparison with other approaches.

Table 6. Content-based evaluation results for summarization schemes.

		Content-based performance parameters							
				Readability					
Datasets	Methods	Cohesion	Non-redundancy	*GFI*	*CLI*	*FKGL*	*ARI*	*SMOG*	*RPS*
DUC 2006	TLBO	**0.683**	**0.9573**	**14.65**	**9.78**	**10.70**	**11.64**	**12.37**	**8.78**
	PSO–GA	0.579	0.9328	13.82	8.48	9.56	10.91	10.42	7.46
	CSO	0.531	0.9217	12.48	8.31	9.22	10.11	10.24	7.13
DUC 2007	TLBO	**0.645**	**0.9547**	**14.28**	**9.43**	**10.41**	**11.41**	**12.20**	**8.53**
	PSO–GA	0.527	0.9231	13.51	8.32	9.36	10.30	10.52	7.21
	CSO	0.506	0.9178	12.11	8.15	9.16	10.06	10.09	7.02

Bold values indicate the overall best performance.



(**a**) Co-selection scores for DUC 2006.

(**b**) Co-selection scores for DUC 2007.

Figure 5. Co-selection scores for the summarization schemes.

iii. TLBO is very simple in computation in comparison with other metaheuristic approaches.

iv. The proposed scheme is based on finding the optimal combinations of sentence scoring methods that are not adopted in any of the schemes used in comparison. This produces better performance for the proposed scheme.

The performance of the proposed scheme is approximately identical for both datasets. Its working is independent of the data.

6.2 *Content-based analysis*

Table 3 shows the scores of cohesion, non-redundancy and readability of the system summary generated by the schemes based on different metaheuristic approaches. The cohesion and non-redundancy scores show that TLBO performs the best among all approaches. This implies that the contents of the generated summary using TLBO are highly related to each other and non-redundant in comparison with other approaches for both datasets. The readability scores calculated in terms of *GFI*, *CLI*, *FKGL*, *ARI* and *SMOGI* are at word level, which shows the ease of reading of contents on the basis of number of complex words, number of syllables, number of sentences and average number of characters. TLBO again performs the best for all the word level readability functions. *RPS* shows the readability at the sentence level by obtaining the relatedness between two adjacent sentences. The scores of *RPS* show that TLBO performs the best among all. Graphical representations of the content-based evaluation metrics show that the performance of the proposed scheme is not data dependent. Figures 6, 7 and 8 show graphical representation of the content-based results.

6.3 *Manual readability testing*

We have also tested the readability of each system-generated summary manually, where two expert assessors give a rating for each system-generated summary on the formats that are readable, partially readable and non-readable, which are given as follows [38, 44].

1. The summary should be understandable, non-redundant and focused to main topic.

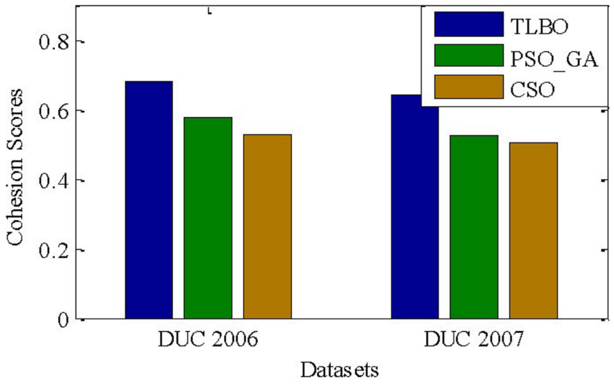


Figure 6. Cohesion scores for the summarization schemes.

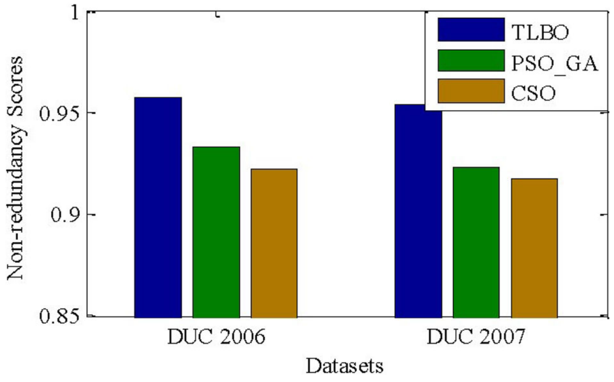
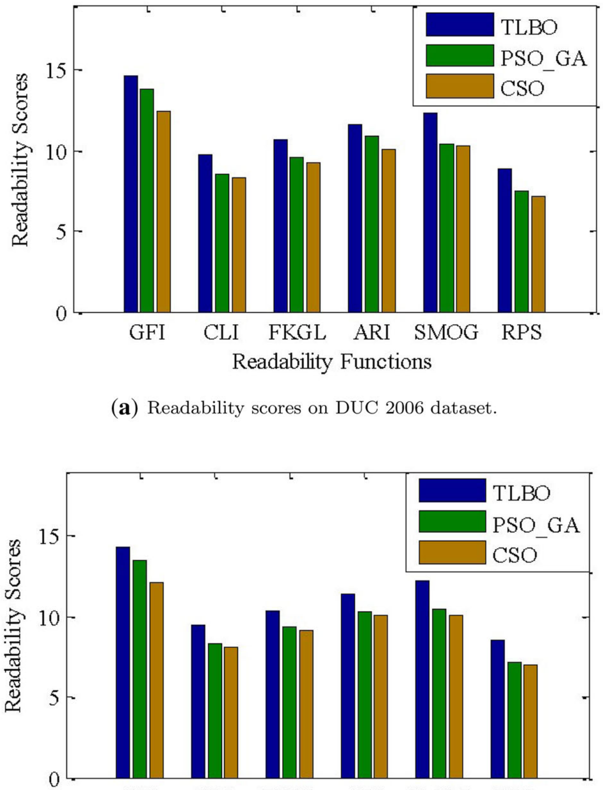


Figure 7. Non-redundancy scores for the summarization schemes.



(**a**) Readability scores on DUC 2006 dataset.

(**b**) Readability scores on DUC 2007 dataset.

Figure 8. Readability scores for the summarization schemes.

2. Sentences of a summary should be complete and linked to each other.
3. The summary should not contain complex sentences.

If a summary follows all of these guidelines, then it is rated as readable. If a summary follows half of these given guidelines, then it is rated as partially readable else it is rated as non-readable [44]. Since human judgments cannot be consistent every time, it is interesting to measure how well two different judges agree on readability. The best way to measure inter-judge agreement is the kappa statistics [45], given as follows:

$$kappa\ (\kappa) = \frac{P(A) - P(E)}{1 - P(E)} \qquad (39)$$

where $P(A)$ is the proportion of the time the judges agreed, and $P(E)$ is the proportion of the time the judges agree by chance. A value of κ-measure in the interval [2/3, 1] is seen as acceptable. The results of manual readability with kappa measures are shown in table 7. The manual readability results confirm that the produced summaries are readable. The results of κ-measure show that all judgments are acceptable.

Table 7. Manual readability testing results for the proposed scheme.

Dataset	Judgment	κ-measure
DUC2006	Readable	0.744
DUC2007	Readable	0.862

7. Conclusion

In this paper, we have discussed a TLBO-based multi-document summarizer to create a generic extractive summary. The proposed summarizer has been compared to hybrid PSO–GA-based summarization scheme and CSO-based summarization scheme. The performance of all the summarizers has been evaluated in terms of precision, recall, F_1, accuracy, cohesion, readability and non-redundancy, on DUC datasets in two stages. The first stage, i.e., co-selection-based evaluation, shows the performance of summarizer on the basis of reference summary, while the second stage, i.e., content-based evaluation, shows the performance of the summarizer on the basis of the contents of the system summary. In most of the cases, TLBO has performed better than other approaches. From the results of good contributed scoring methods, we can say that the derived sentence position method works well. In the future, we intend to examine the proposed scheme with respect to other nature-inspired algorithms.

Nomenclature

Notations	Description
D	document collection
$\|D\|$	number of documents in document collection
d_i	i^{th} document of document collection
$\|d_i\|$	number of sentences in i^{th} document
S	set of sentences in a document
s_j	j^{th} sentence of document
$\|s_j\|$	number of words in j^{th} sentence
w_p	word p of sentence s_j
α_{pj}	*tf–idf* score of a word p in j^{th} sentence
tf	term frequency
idf	inverse document frequency
fw	frequent words
Sum_s	system-generated summary
$Sim(s_j)$	similarity score of a sentence j
X_q	q^{th} solution
New_X_q	updated q^{th} solution

References

[1] Luhn H P 1958 The automatic creation of literature abstracts. *IBM Journal of Research and Development* 2: 159–165

[2] Verma P and Om H 2016 Extraction based text summarization methods on users review data: a comparative study. In: *Proceedings of the Conference on Smart Trends for Information Technology and Computer Communications*. Springer, pp. 346–354

[3] Nenkova A and McKeown K 2012 A survey of text summarization techniques. In: *Mining text data*. Boston, MA: Springer, pp. 43–76

[4] Oliveira H, Ferreira R, Lima R, Lins R D, Freitas F, Riss M and Simske S J 2016 Assessing shallow sentence scoring techniques and combinations for single and multi-document summarization. *Expert Systems with Applications* 65: 68–86

[5] Abbasi-ghalehtaki R, Khotanlou H and Esmaeilpour M 2016 Fuzzy evolutionary cellular learning automata model for text summarization. *Swarm and Evolutionary Computation* 30: 11–26

[6] Alguliev R M, Aliguliyev R M, Hajirahimova M S and Mehdiyev C A 2011 MCMR: maximum coverage and minimum redundant text summarization model. *Expert Systems with Applications* 38: 14514–14522

[7] Asgari H, Masoumi B and Sheijani O S 2014 Automatic text summarization based on multi-agent particle swarm optimization. In: *Proceedings of the Iranian Conference on Intelligent Systems (ICIS)*, IEEE, pp. 1–5

[8] Binwahlan M S, Salim N and Suanmali L 2009 Swarm based text summarization. In: *Proceedings of the Association of Computer Science and Information Technology-Spring Conference (IACSITSC'09)*, IEEE, pp. 145–150

[9] Binwahlan M S, Salim N and Suanmali L 2009 Fuzzy swarm based text summarization. *Journal of Computer Science* 5: 338–346

[10] Binwahlan M S, Salim N and Suanmali L 2010 Fuzzy swarm diversity hybrid model for text summarization. *Information Processing & Management* 46: 571–588

[11] Verma P and Om H 2019 A variable dimension optimization approach for text summarization. In: *Proceedings of the Conference on Harmony Search and Nature Inspired Optimization Algorithms*. Springer, pp. 687–696

[12] Gordon M 1988 Probabilistic and genetic algorithms in document retrieval. *Communications of the ACM* 31: 1208–1218

[13] Khan A, Salim N and Kumar Y J 2015 A framework for multi-document abstractive summarization based on semantic role labeling. *Applied Soft Computing* 30: 737–747

[14] Kogilavani A and Balasubramanie P 2010 Clustering based optimal summary generation using genetic algorithm. In: *Proceedings of the Conference on Communication and Computational Intelligence (INCOCCI)*, IEEE, pp. 324–329

[15] Meena Y K and Gopalani D 2015 Evolutionary algorithms for extractive automatic text summarization. *Procedia Computer Science* 48: 244–249

[16] Shareghi E and Hassanabadi L S 2008 Text summarization with harmony search algorithm-based sentence extraction. In: *Proceedings of the 5th International Conference on Soft Computing as Transdisciplinary Science and Technology*, ACM, pp. 226–231

[17] Rautray R and Balabantaray R C 2017 Cat swarm optimization based evolutionary framework for multi document summarization. *Physica A: Statistical Mechanics and its Applications* 477: 174–186

[18] Rautray R and Balabantaray R C 2017 An evolutionary framework for multi document summarization using Cuckoo search approach: MDSCSA. *Applied Computing and Informatics*. 14: 134–144

[19] Ansamma J, Premjith P S and Wilscy M 2017 Extractive multi-document summarization using population-based multicriteria optimization. *Expert Systems with Applications* 86: 385–397

[20] Verma P and Om H 2019 Collaborative ranking-based text summarization using a metaheuristic approach. In: *Proceedings of the Conference on Emerging Technologies in Data Mining and Information Security*. Springer, pp. 417–426

[21] Nomoto T and Matsumoto Y 2003 The diversity-based approach to open-domain text summarization. *Information Processing & Management* 39(3): 363–389

[22] Jain A and Lobiyal D K 2016 Fuzzy Hindi WordNet and word sense disambiguation using fuzzy graph connectivity measures. *ACM Transactions on Asian and Low-Resource Language Information Processing* 15: 8

[23] Miller G A, Beckwith R, Fellbaum C, Gross D and Miller K J 1990 Introduction to WordNet: an on-line lexical database. *International Journal of Lexicography* 3: 235–244

[24] He Y X, Liu D X, Ji D H, Yang H and Teng C 2006 Msbga: a multi-document summarization system based on genetic algorithm. In: *Proceedings of the Conference on Machine Learning and Cybernetics*, IEEE, pp. 2659–2664

[25] Aliguliyev R M 2009 A new sentence similarity measure and sentence based extractive technique for automatic text summarization. *Expert Systems with Applications* 36: 7764–7772

[26] He R, Qin B and Liu T 2012 A novel approach to update summarization using evolutionary manifold-ranking and spectral clustering. *Expert Systems with Applications* 39: 2375–2384

[27] Alguliev R M, Aliguliyev R M and Isazade N R 2013 Multiple documents summarization based on evolutionary optimization algorithm. *Expert Systems with Applications* 40: 1675–1689

[28] Mendoza M, Bonilla S, Noguera C, Cobos C and Len E 2014 Extractive single-document summarization based on genetic operators and guided local search. *Expert Systems with Applications* 41: 4158–4169

[29] Kusner M, Sun Y, Kolkin N and Weinberger K 2015 From word embeddings to document distances. In: *Proceedings of the Conference on Machine Learning*, pp. 957–966

[30] Kobayashi H, Noguchi M and Yatsuka T 2015 Summarization based on embedding distributions. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1984–1989

[31] Kenter T and De Rijke M 2015 Short text similarity with word embeddings. In: *Proceedings of the 24th ACM International Conference on Information and Knowledge Management*, ACM, pp. 1411–1420

[32] Mikolov T, Chen K, Corrado G and Dean J 2013 Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781

[33] Pinter Y, Guthrie R and Eisenstein J 2017 Mimicking word embeddings using subword RNNs. arXiv preprint arXiv:1707.06961

[34] Rao R V, Savsani V J and Vakharia D P 2011 Teaching–learning-based optimization: a novel method for constrained

mechanical design optimization problems. *Computer-Aided Design* 43: 303–315

[35] Rubner Y, Tomasi C and Guibas L J 2000 The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision* 40: 99–121

[36] Parveen D, Mesgar M and Strube M 2016 Generating coherent summaries of scientific articles using coherence patterns. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 772–783

[37] Sankar K and Sobha L 2009 An approach to text summarization. In: *Proceedings of the Third International Workshop on Cross Lingual Information Access: Addressing the Information Need of Multilingual Societies*, ACL, pp. 53–60

[38] Verma P and Om H 2019 MCRMR: Maximum coverage and relevancy with minimal redundancy based multi-document summarization. *Expert Systems with Applications.* 120: 43–56

[39] Willett P 2006 The Porter stemming algorithm: then and now. *Program* 40: 219–223

[40] Bird S and Loper E 2004 NLTK: the natural language toolkit. In: *Proceedings of the ACL 2004 on Interactive Poster and Demonstration Sessions*, ACL, p. 31

[41] Schlkopf B, Weston J, Eskin E, Leslie C and Noble W S 2002 A kernel approach for learning from almost orthogonal patterns. In: *Proceedings of the European Conference on Machine Learning*. Berlin–Heidelberg: Springer, pp. 511–528

[42] Lin CY 2004 Rouge: a package for automatic evaluation of summaries. In: *Text Summarization Branches Out*

[43] Stajner S, Evans R, Orasan C and Mitkov R 2012 What can readability measures really tell us about text complexity. In: *Proceedings of the Workshop on Natural Language Processing for Improving Textual Accessibility*, pp. 14–22

[44] William H D 2004 *The principles of readability*. ERIC, Online Submission

[45] Ray R L 2010 Introduction to information retrieval. *Journal of the American Society for Information Science and Technology* 4: 852–885