

Research Article

Use of Genetic Algorithm for Cohesive Summary Extraction to Assist Reading Difficulties

K. Nandhini and S. R. Balasundaram

Department of Computer Applications, National Institute of Technology, Tiruchirappalli 620015, India

Correspondence should be addressed to K. Nandhini; nandhnikumaresh@yahoo.co.in

Received 4 March 2013; Revised 30 April 2013; Accepted 23 May 2013

Academic Editor: Baoding Liu

Copyright © 2013 K. Nandhini and S. R. Balasundaram. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Learners with reading difficulties normally face significant challenges in understanding the text-based learning materials. In this regard, there is a need for an assistive summary to help such learners to approach the learning documents with minimal difficulty. An important issue in extractive summarization is to extract cohesive summary from the text. Existing summarization approaches focus mostly on informative sentences rather than cohesive sentences. We considered several existing features, including sentence location, cardinality, title similarity, and keywords to extract important sentences. Moreover, learner-dependent readability-related features such as average sentence length, percentage of trigger words, percentage of polysyllabic words, and percentage of noun entity occurrences are considered for the summarization purpose. The objective of this work is to extract the optimal combination of sentences that increase readability through sentence cohesion using genetic algorithm. The results show that the summary extraction using our proposed approach performs better in *F*-measure, readability, and cohesion than the baseline approach (lead) and the corpus-based approach. The task-based evaluation shows the effect of summary assistive reading in enhancing readability on reading difficulties.

1. Introduction

Students with learning difficulties are diverse in nature, and the difficulties include reading difficulties, math difficulties, writing difficulties, and other problems related with the learning process. About 8% to 10% of students are having learning difficulties [1]. As reading is an essential component in life, our focus is mainly on the learners with reading difficulties. Challenges do exist while comprehending and organizing information from text for such learners. Learners find greater difficulty when they are reading at their earlier stage because of the amount of information to be read [2]. Summarizing can either be taught [3] or understood as one of the several strategies [4] and can be shown to improve their comprehension and recall skills about whatever is read [5].

The impact of summarization in comprehending the text is significant for learners with reading difficulties [2]. As a strategy performed either during or after reading, summarizing helps readers to focus on the main ideas and other key

skill concepts that have been taught and to disregard the less relevant ones [6]. Reading comprehension is defined [7] as “the process of simultaneously extracting and constructing meaning.” This definition emphasizes on both decoding and integrating new meanings with old information. The act of comprehending entails three essential elements, namely, (1) the reader, (2) the text, and (3) the activity [8]. For learners with reading difficulty, the activity element (summarization) can be used to improve reading comprehension. When the learner does not have significant effect under remedial course, the alternate option is to support the learner through assistive tool. As the learner may have a problem in encoding the text, due to slower encoding speed, units could get lost from the working memory before they are being processed [9]. Such learners should map the content of the text to an image, so the trigger words could often create confusion to these individuals [10]. The above-mentioned factor has to be considered while extracting summary sentences. Our objective is to extract summary of a given text that

balances F -measure, readability, and cohesion using genetic algorithm. The learner will be benefited by the summary of the given text which reduces their difficulty in reading and understanding.

Automatic document summarization takes a source document (or source documents) as input, and then it extracts the essence of the source(s), to present a well-formed summary to the user [11]. Document summaries can be abstracts or extracts. An extract summary consists of sentences extracted from a document while an abstract summary may contain words and phrases which do not exist in the original document [11]. The sentences are scored using extracted features which are based on importance and readability. Sentences having higher informative score are extracted and ordered chronologically. The major issue of extractive summarization is the lack of cohesion. Cohesion is the extent to which the ideas in the text are expressed clearly and relate to one another in a systematic fashion, by avoiding a confusing jumble of information. Several studies have clearly established a link between cohesion and comprehension [12]. Higher cohesion has been found to facilitate comprehension and recall. The optimal level of cohesion depends on the knowledge level of the reader [13], and it was found that low knowledge readers could be benefited from high cohesion texts, as most researchers would expect. The objective of this work is to improve cohesion in the extracted sentences to form summary of a specified length by using genetic algorithm.

The rest of the paper is organized as follows. In Section 2, we discuss the related works in the text summarization using genetic algorithm and cohesive summary extraction. Section 3 describes the proposed methodology. In Section 4, the results based on intrinsic evaluation are discussed, and the final section deals with discussions on extrinsic evaluation.

2. Related Works

Text summarization based on machine learning has got tremendous improvement over wide areas of applications. Machine learning algorithms can be applied in almost all the phases of text summarization such as feature selection, finding the optimal weight for each feature, and sentence extraction. Genetic algorithms (GAs) are a relatively new paradigm for a search, based on principles of natural selection. GAs, introduced by Holland, were proven to be the most powerful optimization technique in a large solution space [14, 15]. They are used where exhaustive search for solution is expensive in terms of computation time. The dominance of GAs in each phase of text summarization can be widely seen in various research areas.

In general, Vafaie and De Jong [16] suggested that GAs are a useful tool for solving difficult feature selection problems where both the size of the feature set and the performance of the underlying system are important design considerations. In case of text summarization, Silla et al. [17] investigated the effectiveness of genetic algorithm-based attribute selection to improve the performance of classification algorithms by solving automatic text summarization task. Also, Suanmali et al. [18] proposed an approach called feature fusion technique

to discover which features out of the available ones are most useful for extracting summary sentences.

To determine the feature weights, Litvak et al. [19] designed a new approach to a multilingual single-document extractive summarization using a genetic algorithm which will find an optimal weighted linear combination of 31 statistical sentence scoring methods with all language-independent features. Yeh et al. [20] proposed a novel trainable summarizer, which has several kinds of document features, and two major ideas were employed to improve the conventional corpus-based text summarization. First, sentence positions are ranked to emphasize the significances of different sentence positions; second, the score function was trained by the genetic algorithm to obtain a suitable combination of feature weights. Fattah and Ren [21] investigated the effect of each sentence feature on the summarization task and used all features in combination to train genetic algorithm and mathematical regression (MR) models to obtain a suitable combination of feature weights.

To extract the summary sentences, Fatma et al. [22] and Liu et al. [23] used GAs, to find sets of sentences that maximize summary quality metrics, starting from a random selection of sentences as the initial population. To choose the best summary, multiple candidates were generated and evaluated for each document. Qazvinian et al. [24] used GA to produce document summary. They proposed a fitness function based on following three factors: readability factor (RF), cohesion factor (CF), and topic-relation factor (TRF). Lack of cohesion is the major drawback of extractive summarization, which can be improved by various techniques.

Various attempts were made to improve cohesion in extractive summarization. Smith et al. [25] investigated four different ways of creating 100 word summaries, and the results proved that the summary produced by a traditional vector space-based summarizer is not less cohesive than a summary created by taking the most important sentences from the summarizer. The simplest form of cohesion is lexical cohesion. Morris and Hirst [26] introduced the concept of lexical chains. Researchers explored various types of cohesion measures, namely, clue words that are based on stems, semantic distance based on WordNet, and cosine similarity that is based on the word TF-idf vector. Mani and Bloedorn [27] introduced few methods based on text cohesion which models text in terms of relations between words or referring expressions, to help determine how tightly connected the text is. Silber and McCoy [28] discussed that the lexical chains represent the lexical cohesion among the arbitrary number of related words. By using lexical chains, they can find statistically the most important concepts by looking at the structure in the document rather than the deep semantic meaning. Barzilay and Elhadad [29] presented a less greedy algorithm that constructs all possible interpretations of the source text using lexical chains. Their algorithm selects the interpretation with the strongest cohesion, and then the strong chains are used to generate a summary of the original.

In this direction, we exploit the strength of genetic algorithm for improving the readability through optimal combination of sentences based on lexical cohesion, where the number of sentences depends on the compression ratio.

3. Proposed Work

For a given text, the input text is preprocessed, and features like average sentence length (f1), percentage of trigger words (f8), title to sentence similarity (f2), centrality (f3), percentage of polysyllabic words (f4), percentage of noun occurrences (f5), sentence position (f7), positive keywords (f9), and term frequency (f6) are extracted from the text. The significance of learner-dependent features in summary extraction is discussed in [30]. The sentence score is calculated using weights and features vectors. The sentence combination for summary generation is identified using GA based on sentence similarity and sentence score. The working model of the process is shown in Figure 1. When the input text is large, the number of candidate summary sentences is also large and the feasible combination to form an extractive summary thus grows exponentially. So, the search for optimal solutions under multiple criteria has been proven to be computationally demanding and time consuming. Evolutionary computation has been proven to be an effective way to solve such problems. Genetic algorithm, an evolutionary algorithm, has the ability of avoiding the local search and can increase the probability of finding the global best. The proposed model formulates the sentence extraction problem, that considers both informative score and cohesion using genetic algorithm. An effective integer coding strategy, which can reduce encoding and decoding time, is applied in this process.

3.1. Document Representation. Documents to be summarized can be represented as graph $G = (N, E)$. Sentences in the documents are represented as nodes (N), and the similarity between sentences is represented as edges (E). The graph representation of a sample document is shown in Figure 2. Graph-based extractive summarization algorithm aims at identifying the most important sentences from a text, based on information exclusively drawn from text itself [31]. Graph-based methods are unsupervised and rely on the given texts to derive an extractive summary. The proposed algorithm consists of following steps.

- (1) The informative score of the sentences are estimated using feature vectors.
- (2) Then, similarity between the sentences is estimated using cosine similarity.
- (3) The optimal combination of sentences for subgraph extraction is found, which balances both informative scores and readability using genetic algorithm.

3.2. Informative Score. The Document D consists of a set of sentences $D = \{S_1, S_2, S_3, \dots, S_n\}$. The informative score can be calculated using feature vectors. where the features vectors of the document can be represented using Vector space model. The vector representation of a given document is shown in Table 1.

The informative score can be calculated by linear combination of all features. Where the impact of features varies from domain to domain, the feature weights can be calculated using the linear regression model [32]. Mathematical regression is a good model to estimate the feature weights. In this

TABLE 1: Vector space representation of a document.

Sentences	Features					Class
	F_1	F_2	F_3	\dots	F_m	
S_1	x_{11}	x_{12}	x_{13}	\dots	x_{1m}	y_1
S_2	x_{21}	x_{22}	x_{23}	\dots	x_{2m}	y_0
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
S_n	x_{n1}	x_{n2}	x_{n3}	\dots	x_{nm}	y_1

model a mathematical function relates output to input as in (2):

$$[X] * [W] = [Y]. \quad (1)$$

Here, X is the input matrix (feature parameters). Y is the output vector. W is the linear statistical model of the system (weights W_1, \dots, W_n) in (2):

$$\begin{bmatrix} x_{11} & x_{12} & x_{13} & \dots & x_{19} \\ x_{21} & x_{22} & x_{23} & \dots & x_{29} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{n1} & x_{n2} & x_{n3} & \dots & x_{n9} \end{bmatrix} \times \begin{bmatrix} W_1 \\ W_2 \\ \vdots \\ W_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad (2)$$

where x_{ij} is the value of j th feature in i th sentence and W_j is the weight of feature “ j .” The sentence score can be calculated as a linear combination of the feature vector and its corresponding weight:

$$\text{Score} = \sum_{i=1}^n (W_i * F_i). \quad (3)$$

3.3. Similarity Score. Similarity between the sentences can be calculated using cosine similarity. The terms (t) in the sentences are weighted using Tf-isf as in (6). The term frequency in the given document is the number of times a given term appears in the document:

$$\text{Tf}_i = \frac{T_i}{\sum_{i=1}^n T_k}, \quad (4)$$

where T_i is the number of occurrences of the term and T_k is the sum of occurrences of all the terms in the document. The inverse sentence frequency is a measure of the importance of the term:

$$\text{isf} = \log\left(\frac{N}{n_i}\right), \quad (5)$$

where N is the number of sentences in the document and n is the number of sentences containing the significant term. The corresponding weight is, therefore, computed as,

$$W_t = \text{Tf}_{ij} * \text{isf}_i \quad (6)$$

where Tf_{ij} is the term frequency of i th index term in j th sentence and isf_i is of inverse sentence frequency. The

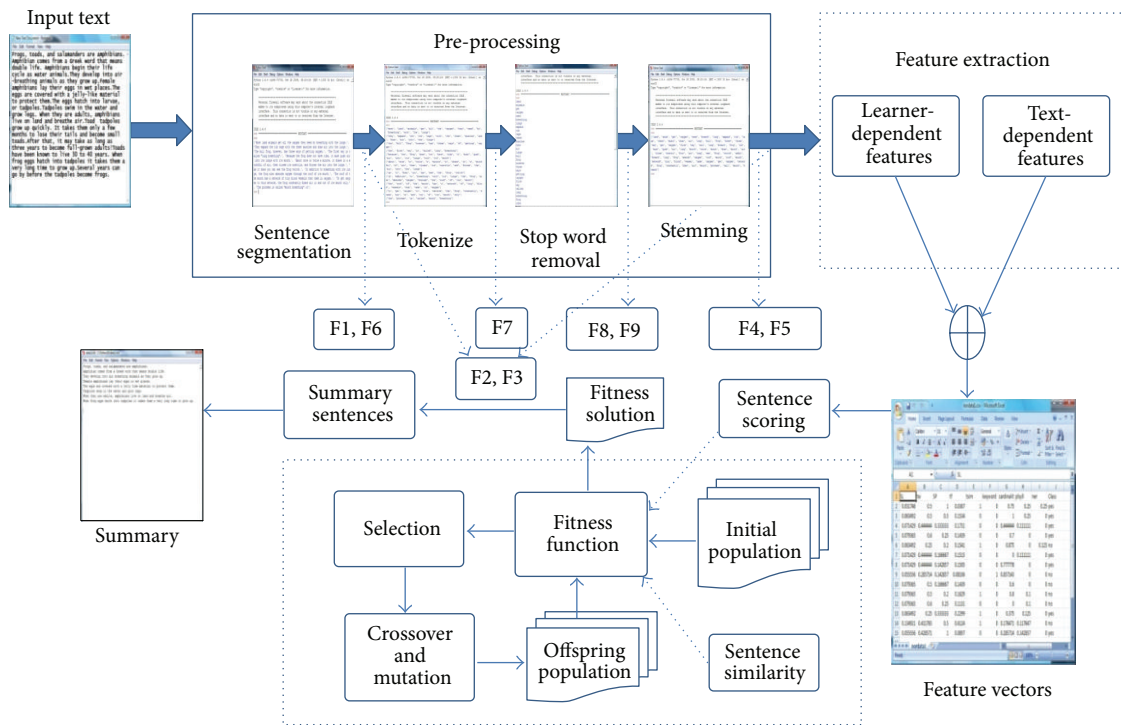


FIGURE 1: Model of the summarization process.

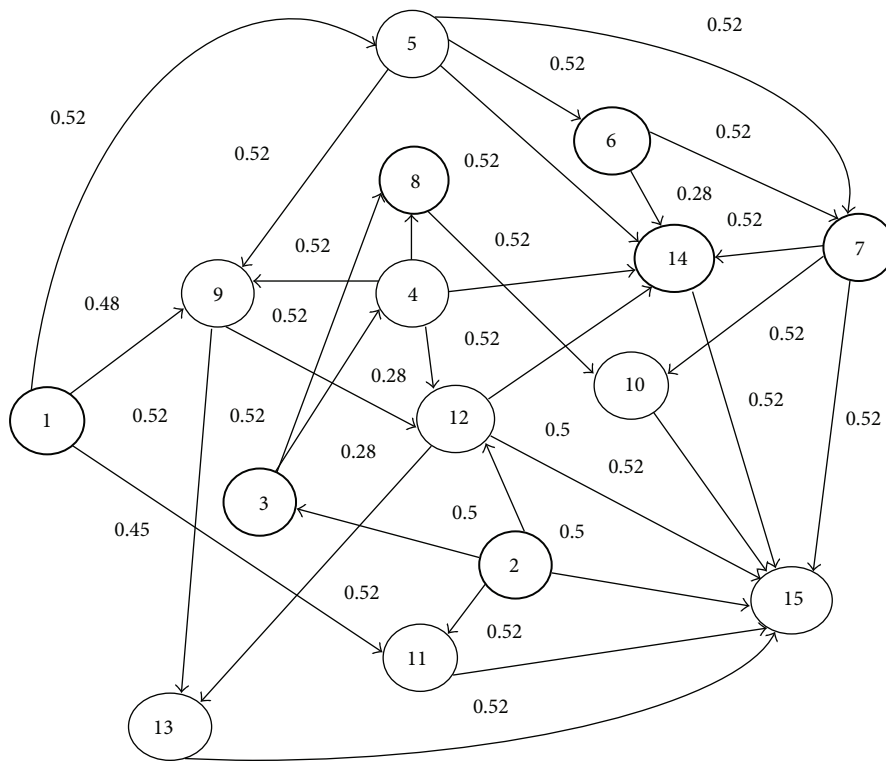


FIGURE 2: Graph representation.

similarity of the sentence can be measured using cosine similarity as in (7):

$$\cos(S_i, S_j) = \frac{S_i \cdot S_j}{\|S_i\| \cdot \|S_j\|} \quad (7)$$

$$= \frac{\sum_{i,j=1}^n S_i * S_j}{\sqrt{\sum_{i=1}^n (S_i)^2} * \sqrt{\sum_{i=1}^n (S_j)^2}} \quad (8)$$

Similarity matrix can be composed using similarity metrics. The lower triangular values are made zero to avoid backward traversal in the graph and to preserve ordering in sentences while extraction, and also the diagonal elements are made zero as there is no self-loop.

3.4. Subgraph Extraction. The subgraph (summary) extraction technique can be formulated as a combinatorial problem. The idea is to extract the high informative score sentences that have good cohesion where the informative score can be calculated using feature score and its weight and cohesion can be measured using the sentence similarity.

3.5. Problem Formulation. The objective of this work is to maximize the average similarity between the sentences having higher informative score. The objective function is as follows:

Maximize

$$\sum_{i=1}^n \sum_{j=i+1}^{n-1} \text{Sim}(S_i * S_j) * \text{Score}(S_j) \quad (9)$$

subject to constraints

$$\text{Sim}(S_i, S_j) = \begin{cases} 0 & \text{if } i > j \\ \text{if } \text{Sim} > 0.8, \\ \text{if } \text{Sim} < 0.1, \\ 1, & \text{otherwise.} \end{cases} \quad (10)$$

Here, the constraint ensures only forward traversal in the graph and ensures that the summary has a cohesive nonredundant candidate sentences. The basic steps in genetic algorithm are given in Algorithm 1.

3.5.1. Chromosome Encoding. A chromosome ξ_a of the proposed algorithm consists of sequence of positive integers that represent the sentence number of the original text for the summary. To represent the solutions of the sentence extraction problem, effective population coding strategy must be designed. Usually binary coding strategies are adopted, in which the size of the chromosome is equal to the input text. Whether a particular sentence is selected or not can be decided by “0” or “1.” Though it is simple to implement, when the text size grows, it may be time consuming to generate a summary. To overcome this problem, integer coding is suggested. Here the size of the chromosome is equal to the size of the summary and is not equal to the input text. Each

locus of the chromosome represents the order of sentences in the summary. The size of the chromosome is fixed and is decided by the compression rate (CR) of the text. With respect to the educational domain, the text at the beginning and at the end is considered as more significant. As the text belongs to the educational domain, the genes of the first and last loci are allocated for the first sentence and last sentence of the text, respectively.

The genes of the chromosomes are sorted in ascending order after random generation in order to preserve the chronological order of the summary, which helps in better comprehension.

3.5.2. Initial Population. Initial population is generated using the encoding technique explained in previous section:

$$\xi^t = \begin{bmatrix} \xi_{11} & \xi_{12} & \xi_{13} & \cdots & \xi_{1m} \\ \xi_{21} & \xi_{22} & \xi_{23} & \cdots & \xi_{2m} \\ & & \vdots & & \\ \xi_{n1} & \xi_{n2} & \xi_{n3} & \cdots & \xi_{nm} \end{bmatrix} [PXN], \quad (11)$$

where P is the population size, N is the size of the chromosome, and t is the number of generations. At random generation “ t ,” any individual chromosome can be represented as

$$\xi_a = [\xi_{(a,1)} \ \xi_{(a,2)} \ \xi_{(a,3)} \ \cdots \ \xi_{(a,n)}]. \quad (12)$$

3.5.3. Fitness Function. After initialization of the population, the process of evaluating the quality of the chromosome is done using the fitness function. The fitness function aids in finding the optimal combination of sentences and tends to balance the informative score and the sentence similarity. The fitness function is as follows:

$$F_a = \sum_{i=1}^n \sum_{j=i+1}^{n-1} \text{Sim}(S_i, S_j) * \text{Score}(S_j), \quad (13)$$

where F_a represents the fitness value of the a th chromosome. n represents the size of the chromosome.

3.5.4. Selection. “Survival of the fittest” can be achieved through the selection operator. The improvement in the average quality of the population can be done by considering good quality chromosome and has to be copied for the next generation. In order to maintain the overall optimal combination, always offspring having better fitness will survive over generations.

3.5.5. Modified Crossover. The objective of exploring the search space can be attained using crossover operator. It examines the current solution in order to find the better ones. Crossover in sentence extraction plays the role of exchanging each partial summary of two selected chromosomes in such a way that the offspring produced by the crossover represents single summary. When pair of individuals has been selected for crossover, single-cut crossover is the best candidate because more number of intermediate points for crossover might lead to redundant information. In single-cut

```

(1) procedure SUMMARY EXTRACTION(.txt file)
(2)   Create initialize population   ▷ choose K different combinations of the sentences
(3)   Evaluate initial population   ▷ Calculate the fitness based on (13)
(4)   Select the best chromosomes   ▷ select the chromosomes with current best fitness
(5)   Generate child chromosome    ▷ apply modified crossover and mutation to produce child chromosomes
(6)   Replace ▷ weaker chromosomes are replaced by potential chromosomes
(7)   Stopping criteria ▷ Repeat steps (3–6) until specified max generation is attained
(8)   Output   ▷ Extract the summary sentences from the best chromosome as final solution
(9)   return Candidate Sentences   ▷ Summary sentences
(10) end procedure

```

ALGORITHM 1: Frame work of the genetic algorithm.

```

(1) procedure CROSSOVER ( $\xi_A^t, \xi_B^t$ )   ▷ The crossover of chromosome A and B
(2)   Input:  $\xi_A^t, \xi_B^t$                  ▷ Chromosomes before crossover
(3)   Output:  $\xi_a^t, \xi_b^t$               ▷ Chromosomes after crossover
(4)   while  $i = 1$  do                 ▷ for the entire size of chromosome
(5)     if ( $\xi_A[i] == \xi_B[i]$ ) then   ▷ check the bit positions of chromosomes
(6)        $matchlist \leftarrow i$        ▷ store the values in a list
(7)     else
(8)        $i \leftarrow i + 1$ 
(9)     end if
(10)  end while
(11)   $s \leftarrow \text{Random}(matchlist)$    ▷ Select a random bit position from list
(12)  for  $i \leftarrow 1, s$  do
(13)     $\xi_a^{t+1}[i] = \xi_A^t[i]$        ▷ Copy partial values of parent(A) to child(a)
(14)     $\xi_b^{t+1}[i] = \xi_B^t[i]$        ▷ Copy partial values of parent(B) to child(b)
(15)  end for
(16)  for  $i \leftarrow s, N$  do
(17)     $\xi_a^{t+1}[i] = \xi_B^t[i]$        ▷ Copy remaining values of parent(B) to child(a)
(18)     $\xi_b^{t+1}[i] = \xi_A^t[i]$        ▷ Copy remaining values of parent(A) to child(b)
(19)  end for
(20)  return  $\xi_a^{t+1}, \xi_b^{t+1}$      ▷ Chromosome after crossover
(21) end procedure

```

ALGORITHM 2: Modified Crossover.

crossover, summaries generated by two different parents are combined at a particular point. In general, after crossover, there is a possibility for occurrences of redundant sentences in summary as the problem is basically a combinatorial one. To avoid the redundancy of genes in chromosome, modified crossover is adapted and shown in Algorithm 2. Here, the crossover between two parent chromosomes is possible only if there is a common gene between them at the same position. The crossover probability depends on the degree of randomness in the genes of the parent chromosome.

3.5.6. Modified Mutation. Mutation is a background operator that produces random changes in various chromosomes and maintains diversity in the population. The population undergoes mutation by an actual change or flipping of one of the genes of the candidate chromosomes, which in turn keeps away from local optima [33]. A random gene in the chromosome is selected and replaced with values that do not duplicate sentences. An arbitrary bit position is chosen and is replaced with values nearer to the values of its neighbor. If

such a neighbor does not exist, another end neighbor will be considered for value selection which is given in Algorithm 3. This reduces the probability of big changes in summary fitness, because sentences which are next to each other in chronological order of the text may talk highly about the same idea [24].

3.5.7. Stopping Criteria. The stopping criteria for implementation purpose can be decided as follows:

- (1) when an upper limit on the number of generations is reached or
- (2) the chance of getting changes in the consecutive generation is extremely low.

3.6. Experiments and Results

3.6.1. Dataset. A collection of hundred and fourteen articles from educational text of grade four to grade seven are used for evaluation. The texts are mainly from science and

TABLE 2: Statistics of the dataset.

Parameter	Size
Number of docs	114
Average number of sent. per doc.	18
Maximum number of sent. per doc.	38
Minimum number of sent. per doc.	11
Summary as % of doc. length	30%
Average summary size (in number of sent.)	7
Maximum number of sent. per summary	12
Minimum number of sent. per summary	5

social subjects because of the challenges faced by the reading difficulties in understanding the concepts. The text basically belongs to three different corpus, one is from the text books [34] and another is of grade level text from Scales [35]. The statistics of the data corpus used in the experiments are tabulated in Table 2.

The summaries of these texts are created manually by three independent human annotators. The sentences to be included in the summary are ranked by the annotators. The sentences that score maximum rank would be included in the summary according to the compression ratio.

3.7. Evaluation. Extractive summary can be evaluated using various characteristics such as accuracy, cohesion, and readability. Accuracy in extraction measures how far the technique is capable of predicting the correct sentence. Evaluation can be classified into intrinsic and extrinsic evaluation. Intrinsic evaluation judges the summary quality by its coverage between machine-generated summary and human-generated summary. Extrinsic evaluation focuses mainly on the quality by its effect on other tasks. We consider both methods because the generated summary should be informative as well as readable. The former part is objective and can be verified using intrinsic evaluation, and the latter part is subjective and can be evaluated using the extrinsic method. In intrinsic evaluation, precision (P), recall (r), and F -measure (F) are used to judge the coverage between the manual and the machine generated summary:

$$\begin{aligned}
 P &= \frac{|S \cap T|}{|S|}, \\
 R &= \frac{|S \cap T|}{|T|}, \\
 F &= \frac{|2 * P * R|}{|R + P|},
 \end{aligned} \tag{14}$$

where S is the machine generated summary and T is the manual summary. The readability of the summary can be evaluated by existing metrics. Extrinsic evaluation can be done to verify the usability of the summary, by its target audience.

3.7.1. F -Measure. F -measure is used to evaluate the extracted summary, as it is a combination of both precision and recall.

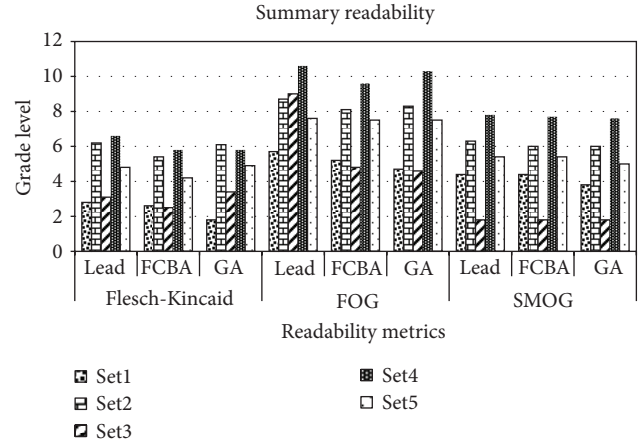


FIGURE 3: Comparison of various summarization methods based on cohesion.

The results of various summarization methods based on the F -measure are tabulated in Table 3. When the compression ratio is 10%, GA performs better in summary extraction when compared to modified corpus-based approach [30] and lead method. When the compression ratio is 20%, it is better than modified corpus approach with a slight improvement, but when 30%, the performance is better than the lead method and lower than the modified corpus-based approach. The reason is that the objective of this approach is not to improve the accuracy of the extraction but to improve its readability or understandability through cohesion, and at the same time is not to dilute the accuracy for the sake of readability. The evaluation criterion focuses mainly on readability and cohesion of the summary.

3.7.2. Readability. To date, there exist numerous readability metrics for predicting the readability of a given text. We considered FOG [36], SMOG [37], and Flesch-Kincaid [38] for predicting the readability of the summary.

The basic difficulty of the text can be measured using average sentence length and average word length. FOG and SMOG predict the readability using average sentence length and the percentage of words with at least three syllables as parameters. The Flesch-Kincaid grade level score uses a fixed linear combination of average words per sentence and average syllable per word. Combining both features result in overall accuracy. Flesch-Kincaid predicts the readability of the summary generated using FCBA as better when compared to others, while FOG predicts the summary generated by both FCBA and GA as better. The comparison of readability score of summaries generated through Flesch Kincaid, SMOG, and FOG are shown in Figure 3. Depending upon the application and the features that have to be considered, suitable readability metrics can be used. For independent-level reading, SMOG works better than the other formulae. It is intended to measure the readability of the material that the teacher has suggested to a student to use independently [39].

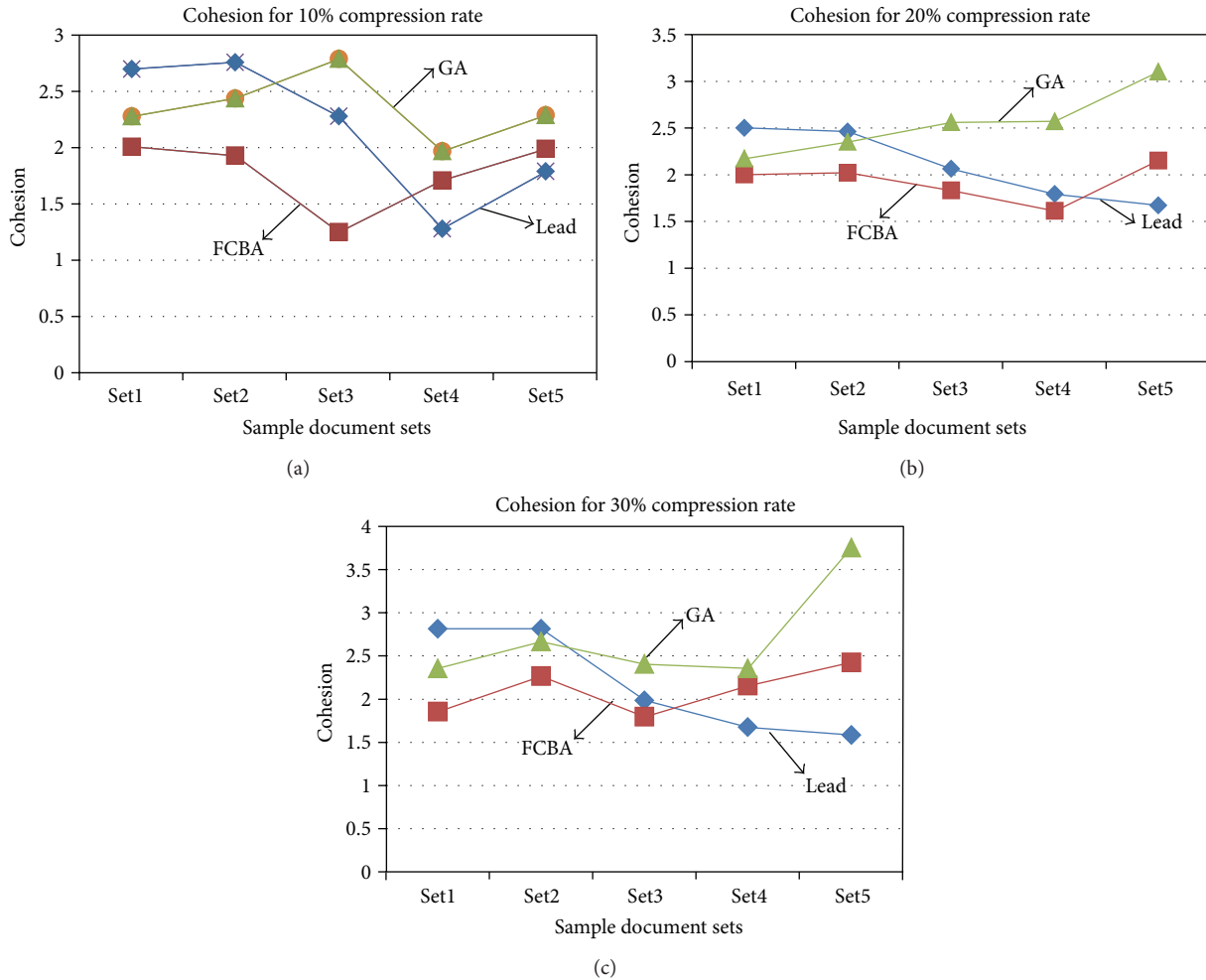


FIGURE 4: Comparison of various summarization methods based on readability.

SMOG predicts that the summary generated by GA produces better readability. The variations are due to the difference in the parameter that metrics consider for readability prediction.

Despite their strengths, there are theoretical shortcomings in using readability formulae to evaluate texts. In particular, the emphasis on shallow features like word length and sentence length is difficult to capture the deep, structural properties of the text that reflect cohesion. The cohesion of the extracted sentences can be measured by applying cosine similarity between consecutive sentences. When sentences in the documents are represented as term vectors, the similarity of two sentences corresponds to the correlation between the vectors.

3.7.3. Cohesion. Cohesion is defined as the set of linguistic means available for creating texture [40], that is, the property of a text being an interpretable whole (rather than unconnected sentences). Lexical cohesion between the sentences in a summary can be measured using cosine similarity and is compared to the summary generated using various models.

For 10% compression, baseline and GA perform better in extracting a cohesive summary. For 20% compression, GA

extracts a better cohesive summary than FCBA and baseline. For 30% compression, GA extracts a better summary and the comparison based on cohesion is measured using cumulative sentence similarity as shown in Figure 4. It is clear that the summary cohesion is mainly based on the content distribution in the text. Baseline may be cohesive, but it fails in predicting the better summary sentences. GA considers all factors without losing any of them, namely, *F*-measure, readability, and cohesion, while extracting a summary.

3.7.4. Task-Based Evaluation. To evaluate the satisfaction of the learners about the summary generated from various methods for over viewing the text, task-based evaluation is used. A way to measure a text's difficulty level is to ask the learner various subjective questions in which they must evaluate how easy it is to understand a text. This metacomprehension of text has been measured using Likert scales in educational studies [41]. A sample set of learners have been asked to fill up a questionnaire using a five-point Likert scale as given below.

- (1) Do you think the summary will help you to understand the text better?
- (2) Do you think the summary is easy to read?

TABLE 3: Comparison of various summarization methods based on F -measure.

	10%			20%			30%		
	Lead	FCBA	GA	Lead	FCBA	GA	Lead	FCBA	GA
Set1	0.31	0.28	0.33	0.41	0.48	0.5	0.42	0.71	0.6
Set2	0.31	0.28	0.33	0.39	0.42	0.6	0.43	0.56	0.66
Set3	0.26	0.5	0.4	0.5	0.66	0.5	0.4	0.83	0.66
Set4	0.25	0.33	0.5	0.44	0.54	0.66	0.56	0.66	0.71
Set5	0.4	0.30	0.5	0.4	0.61	0.57	0.5	0.76	0.62
Avg	0.31	0.34	0.41	0.43	0.54	0.57	0.46	0.70	0.65

TABLE 4: t -test for task based evaluation.

		N	M	SD	t	P
Without assistive summary	Exp group	15	3.3	1.45	-0.168	0.43
	Control group	15	3.26	1.35		
With assistive summary	Exp group	15	3.46	.980	-3.166	0.003
	Control group	15	4.33	.523		

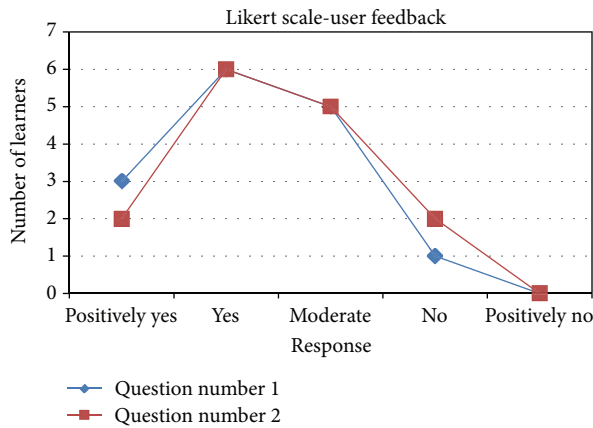


FIGURE 5: User feedback.

The feedbacks obtained from the learners are as shown in Figure 5. User evaluation using Likert scale point is an indirect way of measuring text readability. So, the text readability can be evaluated directly to measure the comprehension of information from a text: objective comprehension questions such as multiple-choice or yes/no type questions, which have been used with learners in educational and psychological studies of comprehension. Though the likert scale feedback is as indirect measurement, the learner may not have clear distinction between the various summaries of the text. The direct way to measure the text comprehension is to evaluate through objective comprehension questions.

The experiments were carried out on fifteen learners with reading difficulties ranging from the fourth grade to the seventh grade. Each learner is given two texts, one from text books with comprehension questions and another with a summary followed by the text and objective questions from the grade level text. The effect of summary assistive reading on students with reading difficulties in enhancing readability and comprehension for both experimental and comparison

groups responses is as shown in Table 4. To analyze the results, paired t -test is used to compare the mean scores and the significance level of experimental and comparison groups with assistive summary and without assistive summary on comprehension test. There was no significant difference in scores between the two groups, when only the text is provided for comprehension test ($t = -0.168$, $P = 0.43$). On the other hand, the groups supplemented with the assistive summary test showed a significant difference between the groups ($t = -3.16$, $P = 0.003$). The experiment group ($M = 4.33$) scored high when compared to the comparison group. From the results, it is evident that an assistive summary has a positive effect in enhancing readability and comprehension of a given text. The learner is given easy to read important sentences which increases his/her interest and motivation. The objective questions focus much on evaluating recall, recognize, and on analysis or comprehension type questions of Blooms taxonomy. It is clear from the questions answered by the learners that the summary helps them in answering literal questions better than analytical questions. Moreover the summary that is easy for one learner may not be easy for another. In future, we plan to develop a graded automatic summarization with various levels that can be better suited to a specific range of learners.

4. Conclusion

When the target audience are specific, the usability of summarization can be measured through task-based evaluation. It is equally important that the quality of the summary is not diluted for its easiness. In this paper, we proposed a genetic algorithm-based text summarization that considers informative score, readability score, and sentence similarity for summary extraction. The performance of GA-based summarization is better in F -measure, readability score and cohesion when compared to corpus-based approach and baseline method (lead). GA-based summary extraction approach can be used to extract summary from the text that belongs to

(1) procedure MUTATION (ξ_R^t)	▷ Mutation in chromosome R
(2) Input: ξ_R^t	▷ Chromosomes before mutation
(3) Output: ξ_r^t	▷ Chromosomes after mutation
(4) $Pos \leftarrow \text{Random}[1 : N]$	▷ select the mutation bit position
(5) Generate R_{num}	▷ generate random number such that
(6) if ($R_{num} \geq \xi_A^t[R_{pos} - 1]$) then	▷ greater than or equal to specified value
(7) if ($R_{num} \leq \xi_A^t[R_{pos} + 1]$) then	▷ less than specified value
(8) if ($R_{num} \neq \xi_A^t[R_{pos}]$) then	▷ not equal to the same value
(9) $\xi_r^{t+1}[pos] \leftarrow R_{num}$	▷ Replace the value of the bit by X
(10) end if	
(11) end if	
(12) end if	
(13) return ξ_r^{t+1}	▷ Chromosome after mutation
(14) end procedure	

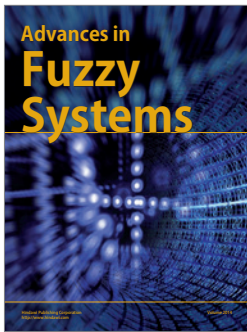
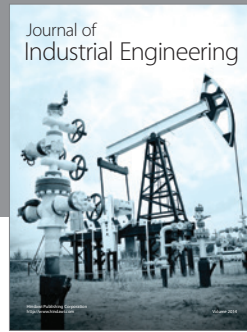
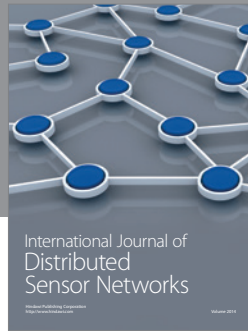
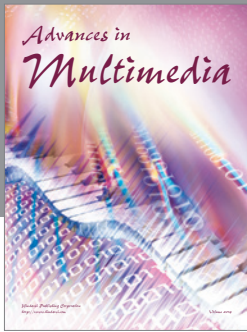
ALGORITHM 3: Modified mutation.

science and social subjects. Furthermore, the summary can be used as an overview before reading the complete text, to improve the readability and comprehension.

References

- [1] International Dyslexia Association, "Definition of dyslexia," *Based in the initial definition of the Research Committee of the Orton Dyslexia Society, former name of the IDA, done in*, 2011, 1994, <http://interdys.org/dyslexiadeinition.htm>.
- [2] M. Gajria, A. K. Jitendra, S. Sood, and G. Sacks, "Improving comprehension of expository text in students with LD: A research synthesis," *Journal of Learning Disabilities*, vol. 40, no. 3, pp. 210–225, 2007.
- [3] B. B. Armbruster, T. H. Anderson, and J. Ostertag, "Does text structure/summarization instruction facilitate learning from expository text?" *Reading Research Quarterly*, vol. 22, no. 3, pp. 331–346, 1987.
- [4] A. S. Palinscar and A. L. Brown, "Reciprocal teaching of comprehension-fostering and comprehension-monitoring activities," *Cognition and Instruction*, vol. 1, no. 2, pp. 117–175, 1984.
- [5] National Reading Panel, *Teaching Children to Read: An Evidence-Based Assessment of the Scientific Research Literature on Reading and Its Implications for Reading Instruction*, National Institute of Child Health and Human Development, National Institutes of Health, 2000.
- [6] M. L. Kamil, "Vocabulary and comprehension instruction: summary and implications of the national reading panel findings," *The Voice of Evidence in Reading Research*, pp. 213–234, 2004.
- [7] A. P. Sweet and C. E. Snow, *Rethinking Reading Comprehension*, Guilford Press, 2003.
- [8] D. R. Reutzel, J. A. Smith, and P. C. Fawson, "An evaluation of two approaches for teaching reading comprehension strategies in the primary years using science information texts," *Early Childhood Research Quarterly*, vol. 20, no. 3, pp. 276–305, 2005.
- [9] P. A. Carpenter and M. Daneman, "Lexical retrieval and error recovery in reading: A model based on eye fixations," *Journal of Verbal Learning and Verbal Behavior*, vol. 20, no. 2, pp. 137–160, 1981.
- [10] R. D. Davis and E. M. Braun, *The Gift of Dyslexia: Why Some of the Smartest People Can't Read... and How They Can Learn*, Penguin Group, 2010.
- [11] I. Mani and M. T. Maybury, *Advances in Automatic Text Summarization*, MIT press, 1999.
- [12] I. L. Beck, M. G. McKeown, G. M. Sinatra, and J. A. Loxterman, "Revising social studies text from a text-processing perspective: evidence of improved comprehensibility," *Reading Research Quarterly*, vol. 26, no. 3, pp. 251–276, 1991.
- [13] D. S. McNamara, "Reading both high-coherence and low-coherence texts: effects of text sequence and prior knowledge," *Canadian Journal of Experimental Psychology*, vol. 55, no. 1, pp. 51–62, 2001.
- [14] J. H. Holland, "Genetic algorithms," *Scientific American*, vol. 267, no. 1, pp. 66–72, 1992.
- [15] L. G. Caldas and L. K. Norford, "A design optimization tool based on a genetic algorithm," *Automation in Construction*, vol. 11, no. 2, pp. 173–184, 2002.
- [16] H. Vafaie and K. De Jong, "Genetic algorithms as a tool for feature selection in machine learning," in *Proceedings of the 4th International Conference on Tools with Artificial Intelligence (TAI '92)*, pp. 200–203, 1992.
- [17] C. N. Silla Jr., G. L. Pappa, A. A. Freitas, and C. A. A. Kaestner, "Automatic text summarization with genetic algorithm-based attribute selection," in *Proceedings of the 9th Ibero-American Conference on Advances in Artificial Intelligence (IBERAMIA '04)*, pp. 305–314, Springer, November 2004.
- [18] L. Suanmali, N. Salim, and M. S. Binwahlan, "Genetic algorithm based sentence extraction for text summarization," *International Journal of Innovative Computing*, vol. 1, no. 1, pp. 1–22, 2011.
- [19] M. Litvak, M. Last, and M. Friedman, "A new approach to improving multilingual summarization using a genetic algorithm," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL '10)*, pp. 927–936, Association for Computational Linguistics, July 2010.
- [20] J.-Y. Yeh, H.-R. Ke, W.-P. Yang, and I.-H. Meng, "Text summarization using a trainable summarizer and latent semantic analysis," *Information Processing & Management*, vol. 41, no. 1, pp. 75–95, 2005.
- [21] M. A. Fattah and F. Ren, "GA, MR, FFNN, PNN and GMM based models for automatic text summarization," *Computer Speech and Language*, vol. 23, no. 1, pp. 126–144, 2009.

- [22] J. K. Fatma, M. Jaoua, L. H. Belguith, and A. B. Hamadou, "Experimentation of two compression strategies for multi-document summarization," in *Proceedings of the International Conference on Computer and Electrical Engineering (ICCEE '09)*, pp. 480–484, December 2009.
- [23] D. Liu, Y. He, D. Ji, and H. Yang, "Genetic algorithm based multi-document summarization," in *PRICAI 2006: Trends in Artificial Intelligence*, vol. 4099 of *Lecture Notes in Computer Science*, pp. 1140–1144, Springer, 2006.
- [24] V. Qazvinian, L. Sharif, and R. Halavati, "Summarizing text with a genetic algorithm-based sentence extraction," *IJKMS*, vol. 4, no. 2, pp. 426–444, 2008.
- [25] C. Smith, H. Danielsson, and A. Jönsson, "Cohesion in automatically created summaries," in *Proceedings of the 4th Swedish Language Technology Conference*, Lund, Sweden, 2012.
- [26] J. Morris and G. Hirst, "Lexical cohesion computed by thesaural relations as an indicator of the structure of text," *Computational Linguistics*, vol. 17, no. 1, pp. 21–48, 1991.
- [27] I. Mani and E. Bloedorn, "Machine learning of generic and user-focused summarization," in *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI '98)*, pp. 821–826, John Wiley & Sons, July 1998.
- [28] H. G. Silber and K. F. McCoy, "Efficient text summarization using lexical chains," in *Proceedings of the 5th International Conference on Intelligent User Interfaces (IUI '00)*, pp. 252–255, ACM, January 2000.
- [29] R. Barzilay and M. Elhadad, "Using lexical chains for text summarization," in *Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization*, vol. 17, pp. 10–17, Madrid, Spain, 1997.
- [30] K. Nandhini and S. R. Balasundaram, "Significance of learner dependent features for improving text readability using extractive summarization," in *Proceedings of IEEE International Conference on Intelligent Human Computer Interaction (IHCI '12)*, 2012.
- [31] N. Kumaresh and B. S. Ramakrishnan, "Graph based single document summarization," in *Data Engineering and Management*, vol. 6411 of *Lecture Notes in Computer Science*, pp. 32–35, Springer, 2012.
- [32] B. Jann, "Making regression tables from stored estimates," *Stata Journal*, vol. 5, no. 3, pp. 288–308, 2005.
- [33] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, 1989.
- [34] <http://www.samacheerkalvi.in>.
- [35] <http://www.impact-information.com/scales.pdf>.
- [36] R. Gunning, "The fog index after twenty years," *Journal of Business Communication*, vol. 6, no. 2, pp. 3–13, 1969.
- [37] G. H. McLaughlin, "Smog grading-a new readability formula," *Journal of Reading*, vol. 12, no. 8, pp. 639–646, 1969.
- [38] J. P. Kincaid, L. R. P. Fishburne Jr., R. L. Rogers, and B. S. Chissom, "Derivation of new readability formulas (automated readability index, fog count and esch reading ease formula) for navy enlisted personnel," 1975.
- [39] J. S. Richardson, R. F. Morgan, and C. E. Fleener, *Reading to Learn in the Content Areas*, Wadsworth Publishing, 2011.
- [40] M. A. K. Halliday and R. Hasan, *Cohesion in english (english language)*, 1976.
- [41] R. H. Maki and S. L. Berry, "Metacomprehension of text material," *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 10, no. 4, pp. 663–679, 1984.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

