applied sciences

MDPI

Article

Improved Text Summarization of News Articles Using GA-HC and PSO-HC

Muhammad Mohsin [1], Shazad Latif [1], Muhammad Haneef [2], Usman Tariq [3], Muhammad Attique Khan [4,*], Sefedine Kadry [5], Hwan-Seung Yong [6] and Jung-In Choi [7]

[1] Department of Computer Science, Shaheed Zulfikar Ali Bhutto Institute of Science and Technology (SZABIST), Islamabad 44001, Pakistan; mmohsin737@hotmail.com (M.M.); dr.shahzad@szabist-isb.edu.pk (S.L.)
[2] Department of Electrical Engineering, Foundation University Islamabad, Rawalpindi 44000, Pakistan; muhammadhaneef@fui.edu.pk
[3] College of Computer Engineering and Science, Prince Sattam Bin Abdulaziz University, Al-Kharaj 11942, Saudi Arabia; u.tariq@psau.edu.sa
[4] Department of Computer Science, HITEC University Taxila, Taxila 47080, Pakistan
[5] Faculty of Applied Computing and Technology, Noroff University College, 4612 Kristiansand, Norway; skadry@gmail.com
[6] Department of Computer Science and Engineering, Ewha Womans University, Seoul 03760, Korea; hsyong@ewha.ac.kr
[7] Department of Applied Artificial Intelligence, Ajou University, Suwon 16499, Korea; jungindb@ajou.ac.kr
* Correspondence: attique.khan@hitecuni.edu.pk

Abstract: Automatic Text Summarization (ATS) is gaining attention because a large volume of data is being generated at an exponential rate. Due to easy internet availability globally, a large amount of data is being generated from social networking websites, news websites and blog websites. Manual summarization is time consuming, and it is difficult to read and summarize a large amount of content. Automatic text summarization is the solution to deal with this problem. This study proposed two automatic text summarization models which are Genetic Algorithm with Hierarchical Clustering (GA-HC) and Particle Swarm Optimization with Hierarchical Clustering (PSO-HC). The proposed models use a word embedding model with Hierarchal Clustering Algorithm to group sentences conveying almost same meaning. Modified GA and adaptive PSO based sentence ranking models are proposed for text summary in news text documents. Simulations are conducted and compared with other understudied algorithms to evaluate the performance of proposed methodology. Simulations results validate the superior performance of the proposed methodology.

Keywords: Automatic Text Summarization (ATS); genetic algorithm; Hierarchical Clustering Technique (HCT); agglomerative clustering; extracted summary; Single Document Summarization

check for updates

Citation: Mohsin, M.; Latif, S.; Haneef, M.; Tariq, U.; Khan, M.A.; Kadry, S.; Yong, H.-S.; Choi, J.-I. Improved Text Summarization of News Articles Using GA-HC and PSO-HC. *Appl. Sci.* **2021**, *11*, 10511. https://doi.org/10.3390/app112210511

Academic Editor: Seungmin Rho

Received: 15 September 2021
Accepted: 5 November 2021
Published: 9 November 2021

1. Introduction

The internet technology known as World Wide Web has seen a lot of advancements in last two decades. In current era internet is cheap and easily available all around the world. This gave rise to exponential growth of information [1]. Due to the presence of large number of users, different kinds of content creating and social networking organizations have turned their direction towards the internet in order to reach a much bigger audience. Even newspapers and news channels have started to adopt internet for news reporting and publishing news articles.

It is cumbersome to read constantly uploaded web pages or articles every minute all around the globe. Moreover, it is not humanly possible to a read large amount of information. Users usually do not read entire web pages or articles, instead users just scan the entire pages or articles just to retrieve few sentences or parts of those sentences to obtain the main crux of the whole information in that article or web page [2]. With such a huge amount of information, it is difficult for the user to identify an important part or parts of sentences which hold the main crux of entire article in less time and with great precision and accuracy [3].

Text summarization is one of the most effective and simplest technique for giving the central idea or main information from the large amount of information. Manual text summarization has been used from early days when the only way to convey information was either through books or through newspapers. The manual text summarization may contain biasness and is time consuming with higher chances of error. Moreover, the accuracy and precision of summary documents not possible for large amount of data [4].

Natural Language Processing (NLP) expresses the interaction between human language and computer. Automatic text summarization is a subfield of NLP and addresses the issue of information retrieval from the data surrounded by redundant information with the help of machine learning. The text summary is generated in fewer amounts of time and with great precision and accuracy [5]. The text summarization with the help of machine learning was proposed by Luhn [3], the model extracted the abstract of the papers and presented them as text summary.

In single document text summarization, a single document is assumed as input and short summarized paragraph is considered as output [6]. Multi-document text summarization was introduced after single document text summarization approach and is more complex than single document text summarization. The multi-document text summarization is similar to single document text summarization. However, the multi-document text summarization takes multiple documents as input and provide single summarized paragraph [7].

Text summarization techniques are also proposed in literature in terms of output. There are two types of text summarization techniques, one is extractive text summarization approach, and the other is abstractive text summarization technique. In extractive text summarization, the final summary considered the same sentences as they are provided in the input document only the important sentences are selected and joined in one paragraph and presented as extractive summary [8]. Extractive text summarization was the first one to be introduced in the domain of automatic text summarization [9]. The second type of approach for automatic text summarization is called abstractive text summarization. Abstractive text summarization uses the same methodology for identification and extraction of sentences but give output summary in different words and sentences conveying the same meaning [10]. Abstractive summarization is like a human writing the summary in its own words rather than using same sentences and words from the document. Abstractive text summarization came after extractive text summarization, and is more complex than extractive text summarization techniques [11]. There are many text summarization techniques are existing in literature. Accuracy of summary is still a challenging issue in text documents. Text summarization is considered as non-convex and NP hard problem. Metaheuristic approaches are excellent in dealing with non-convex and NP hard problems. Therefore, the proposed work considers evolutionary computing approaches to summarize the text document.

The rest of the paper is organized as follow: Section 2 discusses the relevant research work in the domain of text summarization. In Section 3, the proposed technique for extractive text summarization is explained in detail. In Section 4, the experimental setup and simulation results are discussed. Finally, Section 5 concludes the proposed research work.

2. Related Work

The literature review is devided into three subsections according to document type whether it is single document or multidocument.

The authors in [12] discussed k mean clustering for text summarization. Moreover, scores are assigned to cluster on the bases of APRIORI probability. Finally, the sentence with high score is selected for summarization. An automatic extractive text summarization approach using genetic algorithm is proposed in [13] for optimizing the features scores and applied fuzzy logic for score assignment to all the sentences. Finally, high scored sentences are selected and presented in the summary.

The model in [14] considered the text rank algorithm for text summarization. A Cascading Style Sheet (CSS) property for web designers is also introduced which can reduce the lengthy text on smaller screens with the help of text summarization. In [15], word-sentence relation with unsupervised graph ranking is proposed. The model integrates intrinsic value of words and sentences with good accuracy.

In [16], the proposed ensemble model makes use of parallel ensemble approach with classification performed on voting system for text summarization. A bug report text summarization technique is presented in [17]. The model applied fuzzy c-mean clusters for similar sentences and fuzzy logic for making decision of adding or discarding sentences for final summary.

The text summarization technique discussed in [18] investigated semantic and statistical features for summarization of text. The model used Word2Vec for extracting semantics and K-means for grouping similar sentences in addition, ranked all the sentences and top *n* ranked sentences are considered as document summary. However, other clustering techniques such as fuzzy c mean and hierarchal clustering are not considered.

An Arabic single document text summarization model is presented in [19]. The authors presented two text summarization approaches: one is scored based approach and other is binary classifier approach. The binary classifier is trained to predict whether the sentence is a part of final summary or not.

An adaptive and Knowledge-based Event-index (KB-EI) cognitive model is introduced in [20]. The model applied cognitive based process on human memory and emotions for text summarization task. The model has learning phase for identification of information rich sentences and summarization phase for summarizing the document with important sentences.

Multi Document Summarization (MDS):

In [21], authors presented multi-document extractive text as a multi-objective optimization problem and proposed Artificial Bee Colony Optimization (ABC) algorithm to generate text summary. The authors in [22], applied Recursive Neural Network (RNN) for extorting images present in document and employed logistic classifier for finding probability of each sentence present in document for generating the final summary. In [23], PSO is applied on discrete and continues vector space, and sentimental analysis is used for removing redundancy. The model discussed in [24] considered Shark Smell Optimization (SSO) algorithm for summarization of multiple documents. SSO is investigated for optimizing the weights of the features extracted which are used for document summarization.

The authors in [25] presented a text summarization model based on centroid technique and sentence embedding. An abstractive text summarization model is used in [26]. The model uses Generative Adversarial Network (GAN) with time decay attention mechanism for selection of important sentences and summary generation. In [27], the authors proposed 27 rules for classification of text for summary generation using fuzzy logic. The authors in [28] proposed fuzzy logic for identification and mapping of overlapping words. For overcoming duplicate sentence issue in text document, the proposed approach considered graph-based technique for generating summary.

An extractive single document text summarization technique is discussed in [29]. According to authors of the proposed technique, there is no work existing in scientific literature which addresses the text summarization task with semi-graph approach. The proposed technique uses semi-graph approach ESSg for summarizing text. A meta-heuristic optimization model multi-document text summarization approach is discussed in [30]. The approach uses Cat Swarm Optimization (CSO) algorithm for text summarization of multiple-documents. In [31], authors proposed fuzzy logic for text summarization and cosine similarity function is applied for removing redundancy from the extracted summary.

The authors of [32] have discussed a Multi-Modal Summarization (MMS) technique for summarizing text, image, audio, and video. The proposed technique used LexRank algorithm for audio and text summarization and a cross modal analysis is used to bridge the gap between text and images. In videos key frames are extracted. Semantic analysis is performed using pre trained models on Flickr30K and MSCOCO dataset.

Hybrid Document Summarization (HDS):

HDS consists of both single document and multi documents. The authors in [33] introduced a new hierarchical structure based on Recurrent Neural Network (RNN) for extractive text summarization. The model has two levels of attention mechanism applied at word level and at sentence level. A hybrid neural extractive text summarization model known as Contextualized-Representation Hierarchical-Attention Summarization (CRHA-Sum) network is proposed in [34]. The model has ability to learn contextual semantic meaning and features relation for the purpose of text summarization. The model consists of word level attention and sentences level attention. Greedy approach is employed in sentence level attention for selection of sentences.

The text summarization model in [35] used clustering and optimization algorithms for text summarization. The model used K-means algorithm for clustering and an extended version of differential algorithm known as binary differential algorithm for text summarization task. The proposed model is known as COSUM. The text summarization model explained by authors in [36] applied sentence role labeling for semantic analysis and an undirected weighted graph model for text summarization. The model summarizes both single document and multiple documents. The model employed PageRank algorithm for generating graphs. The proposed model is called SRL-ESA-TextSum.

3. Propose Model

Figure 1 gives a pictorial representation of proposed model. The proposed model consists of four stages and one output stage. In stage one, preprocessing is applied on the document for removing inconsistency and for normalizing text. In stage two, the semantics of words are extracted with the help of distributional semantic model. In stage three, a clustering algorithm is applied for making groups of similar sentences in the text. In stage four, optimized ranking algorithm is introduced for assigning ranks to each sentence in the document. Finally, the sentence ranks are normalized, and the summary is generated with given threshold in the output stage.

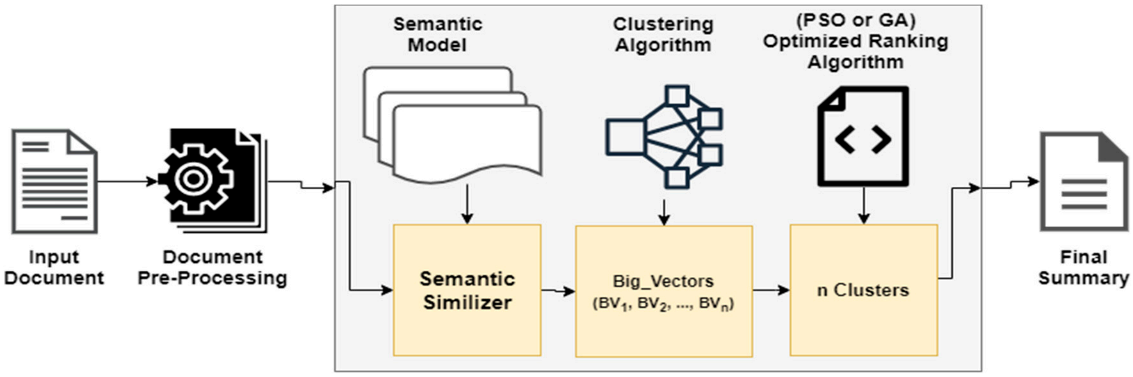


Figure 1. Proposed model overview.

A. Pre-Processing

The purpose of pre-processing stage is to eliminate inconsistency in the data and generate normalize data. For pre-processing, we used Stanford core NLP package [37]. The proposed model pre-possessing steps are as follow.

Removing URL's: URLs existing in document are removed in pre-processing stage. Lower-Case: All the upper-case words are converted to lower-case, present inside the document.

Remove Stop-Words: Stop words do not have any meaning in text summarization. Therefore, the stop-words are removed with the help of list of stop words provided by Stanford core NLP package.

Tokenization: Tokenization is a process of splitting words into individual elements. Each sentence is tokenized into words before passing to the model.

Lemmatization: The tokenized words are further reduced to their root word or stem word.

B. Semantic Extraction

The proposed model applied distributional semantic algorithm for extraction of semantics of words. The distributional semantic model does not depend on lexical and linguistic analysis. The distributional semantic model generates dynamic semantic in high dimensional vector space using statistical analysis of the various kinds of words. The distributional method creates semantic embedding with the help of statistical calculations of the environment in which word exists. The high dimensional real value vector of every word is computed and is presented in word-vector or word embedding form.

Our proposed model applied Word2Vec model [38] based on distributional semantic algorithm for extraction of semantics of words. Word2Vec employed neural network model consisting of two-layers. The model takes text data as input and generates word vectors. Continuous Bag of Words (CBOW) and Skip-Gram are Word2Vec approaches. Skip-Gram predicts context (surrounding words) of word from the word while CBOW predicts word from the context (surrounding words) of the word. The proposed model considered Skip-Gram approach of Word2Vec. In this research work, the Word2Vec model is pre-trained on Google news dataset.

Word2Vec has produced good results in extracting semantics as it has been used in many techniques. Word2Vec is employed in [39] for bootstrapping to generate automatic annotated emotional corpus. Word2Vec is used for solving the words sense in word sense disambiguation in [40]. Word2Vec is applied by [41] for extraction of semantics of words for text coherence problem.

C. Big Vectors

Big vectors represent a sentence's rich semantic content and are based on the distribution hypothesis. It is a semantically comparable bag-of-words representation of a sentence comprising of semantically similar words in particular. These massive vectors of all sentences in a document are obtained. Because the number of words in a phrase varies, big vectors of various sizes are generated. Big vectors of sentences are created by joining similar words together obtained from Word2Vec model. The big vectors of sentences are generated by summing all the words vectors present in the sentences shown in Algorithm 1.

Using above mentioned methodology, big vectors are obtained for each sentence in the document. Since the number of words are not same in all sentences the size of big vectors is fixed to n number of dimensions.

Algorithm 1: Agglomerative Clustering

```
1.  Begin
2.  Initialize with n number of clusters with each cluster containing one element
3.  Calculate the least distance between pair of clusters
4.  Calculate the most 1ike pairs of the clusters
5.  Update the distance matrix
6.  Repeat from step 3 if there are more than one clusters are left
7.  End
```

D. Clustering

In this paper, clustering is used to group similar sentences in the document. Hierarchical clustering is employed in the proposed model.

Hierarchical clustering considers each point as an individual cluster and all points as one cluster. Hierarchical clustering considers distance parameter to combine or divide each cluster. There are two approaches in hierarchical clustering as shown in Figure 2.

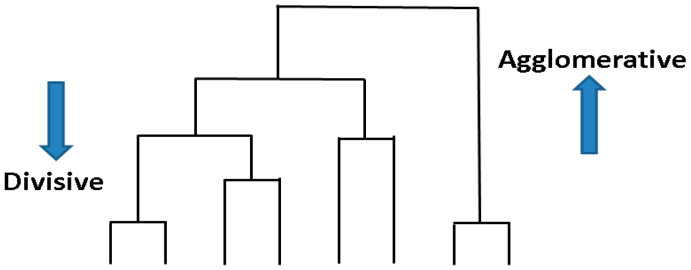


Figure 2. Hierarchical Clustering.

In hierarchical agglomerative clustering (HAC), all points are individual clusters initially. The individual clusters combine to make big clusters on the bases of distance, and this keeps on repeating until all data points are in one big cluster also known as bottom-up approach. On the other hand, the divisive clustering approach (DCA) is opposite to HCA. In (DCA) initially there is one big cluster, and it keeps on dividing clusters until all data points are individual clusters also known as top-bottom approach. Our model used HAC for grouping similar sentences together.

E. Ranking Algorithm

The ranking model employed multiple statistical features for ranking each sentence from the clusters and then these extracted values are normalized. Modified genetic algorithm is used for optimizing the scores of the features and finally the sum of all the optimized features scores constitutes the rank of that sentence. The statistical features used by out model are as follow;

Sentence Length:

According to [42], the length of sentence is connected to the importance of sentence. The proposed model considers sentence length as one of the features for ranking the sentences.

TF-IDF Score:

Term Frequency Inverse Document Frequency (TF-IDF) is one of the most important and widely used feature in entire literature of text summarization. TF-IDF plays an important role in identification of most important words goes through the text document. These words can help in extraction of important sentences form the document. TF-IDF is expressed as

$$s_i^{tf} = \sum_{w \subseteq s_i} t_f\,(w) \tag{1}$$

where s_i^{tf} is ith sentence's sum of all tf-idf score of words in that sentence, $t_f\,(w)$ is a function that returns TF-IDF score of a word w.

Sentence Position:

The most important sentences in the news document are located in the beginning and at the end of the document [42,43]. The position of sentence is calculated as follows:

$$s_i^p = 1 - \frac{s_i - 1}{|S|} \tag{2}$$

where s_i^p is score of sentence position for ith sentence form the given input document and $\mid S \mid$ represent the cardinality of sentences set S of whole document.

Noun Phrases and Verb Phrases:

The sentences which have high number of noun phrases and verb phrases contains good amount of information [44]. We used Stanford POS Tagger for extraction of noun phrases and verb phrases from each sentence in the document.

Proper Noun:

The sentences which contain proper nouns hold a lot of information and is important for text summarization of the document [45].

Aggregate Cosine Similarity:

Cosine similarity is used for computing similarity between two vectors. The work conducted by [46] proposed that cosine similarity effective in important sentence extraction. Many text summarization models proposed in literature use cosine similarity feature for extracting important sentences. The cosine similarity can be calculated as

$$s_i^c = \frac{\sum_{j=1,\quad j\neq i}^{|S|} c\left(s_i, s_j\right)}{|S|} \tag{3}$$

where s_i^c is aggregate cosine similarity score ith sentence, $c\left(s_i, s_j\right)$ indicates cosine similarity between s_i and s_j. System model flow diagram shown in Figure 3.

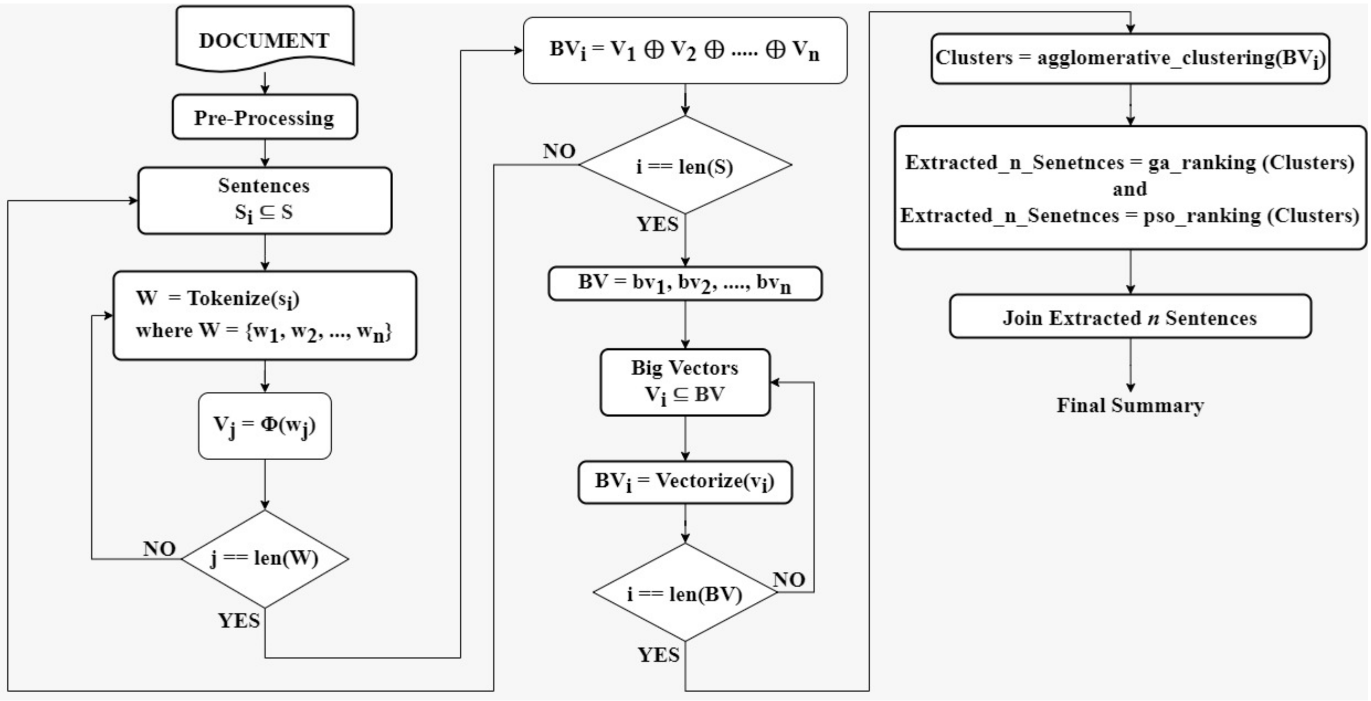

DOCUMENT
Pre-Processing
Sentences
$S_i \subseteq S$
W = Tokenize(s_i)
where W = {w_1, w_2, ..., w_n}
$V_j = \Phi(w_j)$
j == len(W)
NO
YES
$BV_i = V_1 \oplus V_2 \oplus ..... \oplus V_n$
i == len(S)
NO
YES
BV = bv_1, bv_2, ..., bv_n
Big Vectors
$V_i \subseteq BV$
BV_i = Vectorize(v_i)
i == len(BV)
NO
YES
Clusters = agglomerative_clustering(BV_i)
Extracted_n_Senetnces = ga_ranking (Clusters)
and
Extracted_n_Senetnces = pso_ranking (Clusters)
Join Extracted *n* Sentences
Final Summary

Figure 3. Flow diagram of proposed model.

F. Genetic Algorithm (GA)

GA is inspired from the bio evolution of species [47]. Text summarization is considered as NP-hard problem and GA has proved to be effective in solving NP-hard problem. Recently evolutionary algorithms havegained popularity in the domain of text mining and more specifically in text summarization. A lot of text summarization models presented in literature are using GA to optimize the weights of sentences and identification of important sentences. GA is slow in convergence. However, GA is robust and produced best possible optimized results over several generations of population by applying mutation and crossover operators. In this research work, we used non-dominated sorting algorithm-II (NSGA-II) to optimize the rank of sentences. NSGA-II preserved the best individuals of each population and used to create offspring. NSGA-II is fast converging algorithm as compared to conventional GA. Algorithm 2 represents the working of GA. Moreover, the crossover process is illustrated in Figure 4.

Algorithm 2: Genetic Algorithm

```
1.   Begin
2.   Set Parameters
3.   Choose encoded method
4.   Generate initial population
5.   Calculate Fitness value
6.   Perform Selection
7.   Perform Crossover
8.   Perform Mutation
9.   If number of iterations not completed go to Step 5
10.  Decode individual with maximum fitness value
11.  Return best solution
12.  End
```

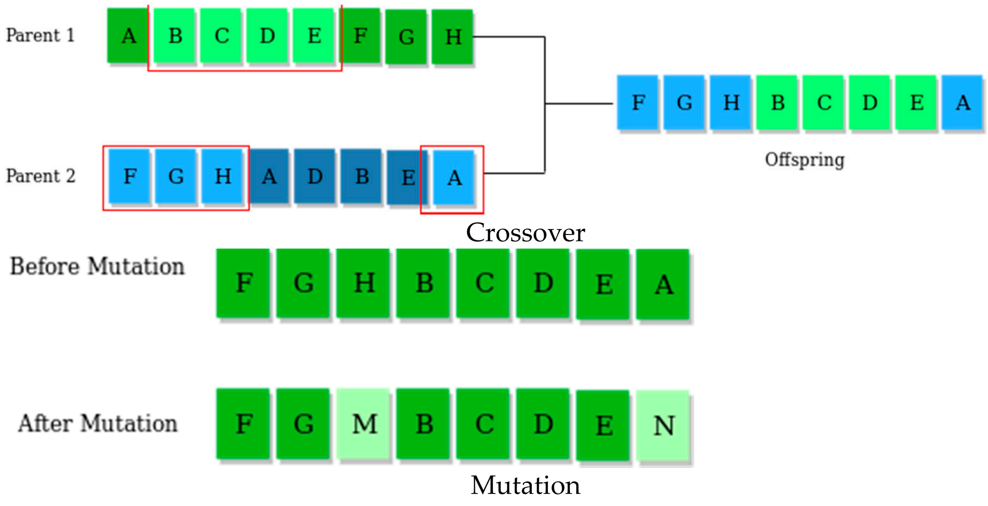


Figure 4. GA crossover and mutation procedure.

G. Particle Swarm Optimization (PSO)

PSO is inspired from the social behavior of bird of flocks [48]. PSO is population based, which is based on the concept of swarm intelligence and is capable of solving NP-hard problems. PSO used personal best and global best experience to update the position of swarms. In PSO, particle represents the candidate solution and particle best position is considered as the optimized solution. The position of the particles depends on velocity of particles and calculated as follows

$$v_k \ = wv_k + c_1 r_1 (pBest_k - d_k) + c_2 r_2 (gBest_k - d_k)$$

$$d_k = d_k + v_k$$

where v_k , d_k represent the velocity and position of kth particle, respectively. w, c_1, r_1, c_2, r_2 are adaptive parameters which are tuned to obtained best possible solution. The working of PSO is given in Algorithm 3.

Algorithm 3: Particle Swarm Optimization Algorithm

```
1.  Begin
2.  Randomly assign initial values to the position and velocity of each kth particle
3.  Compute the fitness of kth particle
4.  Compute $pBests_k$ of kth particle
5.  Compute $nBest_k$ of the entire swarm
6.  Update the $v_k$ velocity of kth particle
7.  Update the $d_k$ position of kth particle
8.  Compute the fitness of kth particle
9.  Update $pBest_k$ of kth particle
10. Update $gBest_k$ of entire swarm
11. End the algorithm if stopping condition met else jump to step 6
12. End
```

H. Complete Ranking Model

First the features are extracted from the sentences in the clusters. The best solution obtained from GA is multiplied with all the feature scores, and, finally, all the features score are added and accumulate result is the rank of that sentence. The proposed PSO-HC and GA-HC is presented in Algorithms 4 and 5.

Algorithm 4: Proposed Model(PSO-HC)

```
1. Begin
2. Let d be the input text document containing S sentences
3. Generate list of Big Vectors
4. Clusters are generated using agglomerative clustering algorithm
5. The clusters are passed to PSO ranking model
6. Final summary is obtained by combining top n selected sentences
7. End
```

Algorithm 5: Proposed Model (GA-HC)

```
1. Begin
2. Let d be the input text document containing S sentences
3. Generate list of Big Vectors
4. Clusters are generated using agglomerative clustering algorithm
5. The clusters are passed to GA ranking mode
6. Final summary is obtained by combining top $n$ selected sentences
7. End
```

4. Experimental Setup and Results

A. Dataset

For evaluating our proposed model, we used Document Understanding Conference (DUC) 2007 dataset and CNN/Daily mail dataset 2015. The dataset contains 10 main topics, and each topic contains four to five sub topics. Each sub topic contains 25 documents. The documents are news articles taken from various news sources. There are four summaries created by experts and two base summaries are considered. In addition, 30 summaries are submitted by the participants.

CNN/Daily Mail is a text summary dataset. Human-made abstractive summary bullets were generated as questions (with one of the elements obscured) and stories as the appropriate passages from which the system is anticipated to answer the fill-in-the-blank question from news stories on CNN and Daily Mail websites. The scripts that crawl, extract, and produce pairs of excerpts and questions from these websites were released by the authors.

The CNN articles were written between April 2007 and April 2015. The Daily Mail articles were written between June 2010 and April 2015. In all, the corpus has 286,817 training pairs, 13,368 validation pairs and 11,487 test pairs, as defined by their scripts. The source documents in the training set have 766 words spanning 29.74 sentences on an average while the summaries consist of 53 words and 3.72 sentences.

B. Pre-Processing

Preprocessing is a process of cleaning the data and shaping the data for the model. Noise is present in the news data that effects the extraction of information, therefore preprocessing is used for cleaning the data from noise. The data preprocess involved in our proposed model is removing URLs and stop words, tokenization of words and sentences, lemmatization of words, extraction of proper nouns, extraction of verb phrases and noun phrases. NLTK tool kit is used for preprocessing the documents.

C. Evaluation Method

We used ROUGE measures for evaluation of proposed model. ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation. ROUGE was proposed by Chin-Yen in 2004 [49]. ROUGE evaluates the summaries generated by the machine learning model with the summaries created by humans. ROUGE evaluates these summaries by counting overlapping basic units such as word sequence, word pairs and $n-$gram grammar between the generated summaries and reference summaries.

D. Comparison with Other Algorithms

To evaluate the performance of proposed algorithm, we considered two clustering algorithms and one optimization algorithm identified from literature that is as follow:

Clustering Algorithms:

The two clustering algorithms are as follow;

K-Means: numerous text summarization models presented in literature used K-Means algorithm [50] for grouping similar sentences from the document.

Fuzzy C-Means: in text summarization models in literature used Fuzzy C-Means [51] in their work for making clusters of similar sentences.

Comparison with Other Methods.

Semantic Text Summarizer [18] model is considered for comparison because same preprocessing is used in [18].

PKUSUMSUM [52] is a Java platform-based summarization model. PKUSUMSUM summarize multiple languages text. The model has the ability to summarize single document, multi-document and topic-based multi document summarization.

TextRank [53] is a graph-based model presented in 2004. It generates a graph of the given text data in which the sentences are represented as a vertex.

OPINOSIS [54] is a graph-based text summarization model which is capable of creating short and concise summaries of the given text document. OPINOSIS generates abstractive summaries of given text documents.

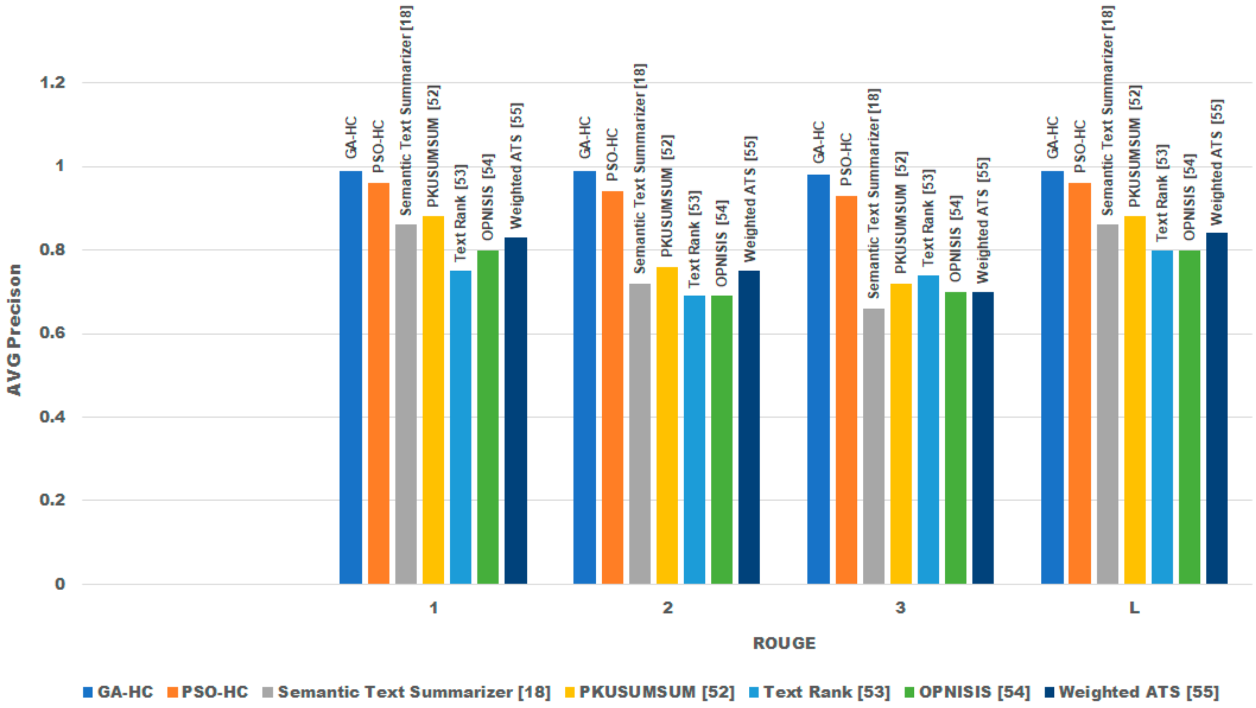
Weighted ATS [55] is a weighted word vector representation method for TF-IDF for ATS. The proposed model is a presumptive way for capturing all possible semantic meanings from text, as well as statistical and linguistic aspects, for large data on the internet. By distinguishing semantically distinct sentences, the proposed word vectors help to improve the diversity of the resulting summary.

Experimental Result

For model evaluation purposes all the documents from the dataset are summarized using proposed models and two understudied models. Summaries are passed to Rouge 1, Rouge 2, Rouge 3, and Rouge L. The average results for each Rouge value (precision, recall, and F1-score) are presented of all the summaries.

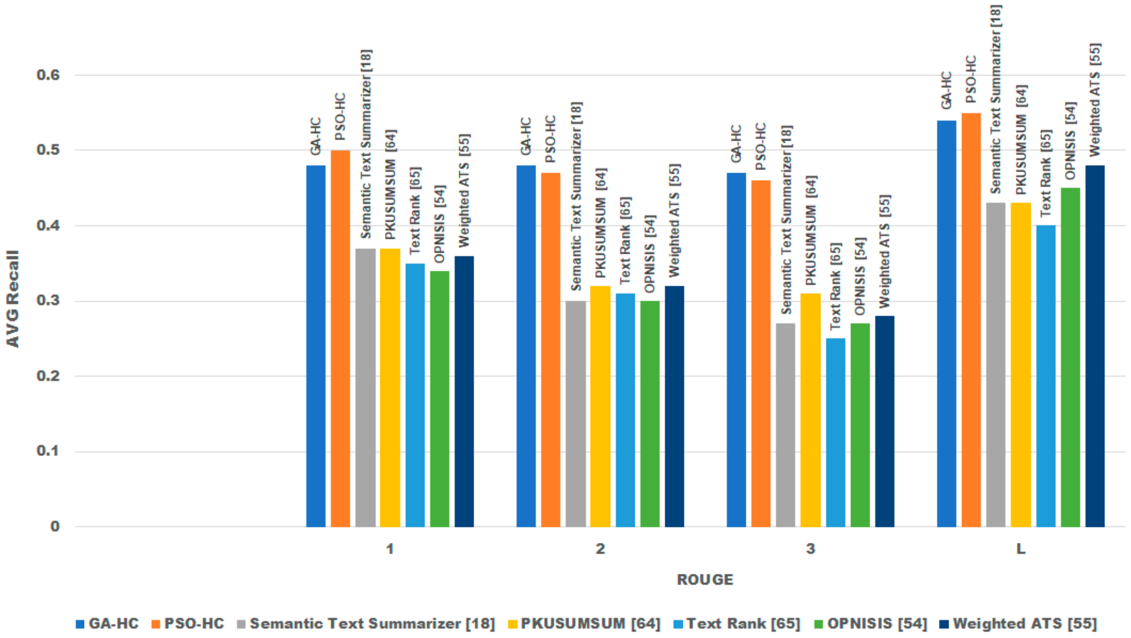
Figure 5a–c represents the average precision, average recall, and F1 score, respectively, for DUC dataset. These figures represent the simulations results of proposed algorithms

GA-HC and PSO-HC with other understudied models. For evaluation purposes, the proposed algorithms GA-HC and PSO-HC are also compared for CNN/Daily mail dataset in Figure 6a–c for average precision, average recall, and F1 score, respectively. The results are compared with understudied algorithms. In all results, proposed algorithms performed efficiently in comparison with other algorithms due to exploration and exploitation capabilities of GA-HC and PSO-HC. Due to exploration capability proposed algorithms easily escape from local optimum values, and due to exploitation capability, achieves better optimum values. The GA-HC performed better than PSO-HC due to more exploration capability than PSO-HC.

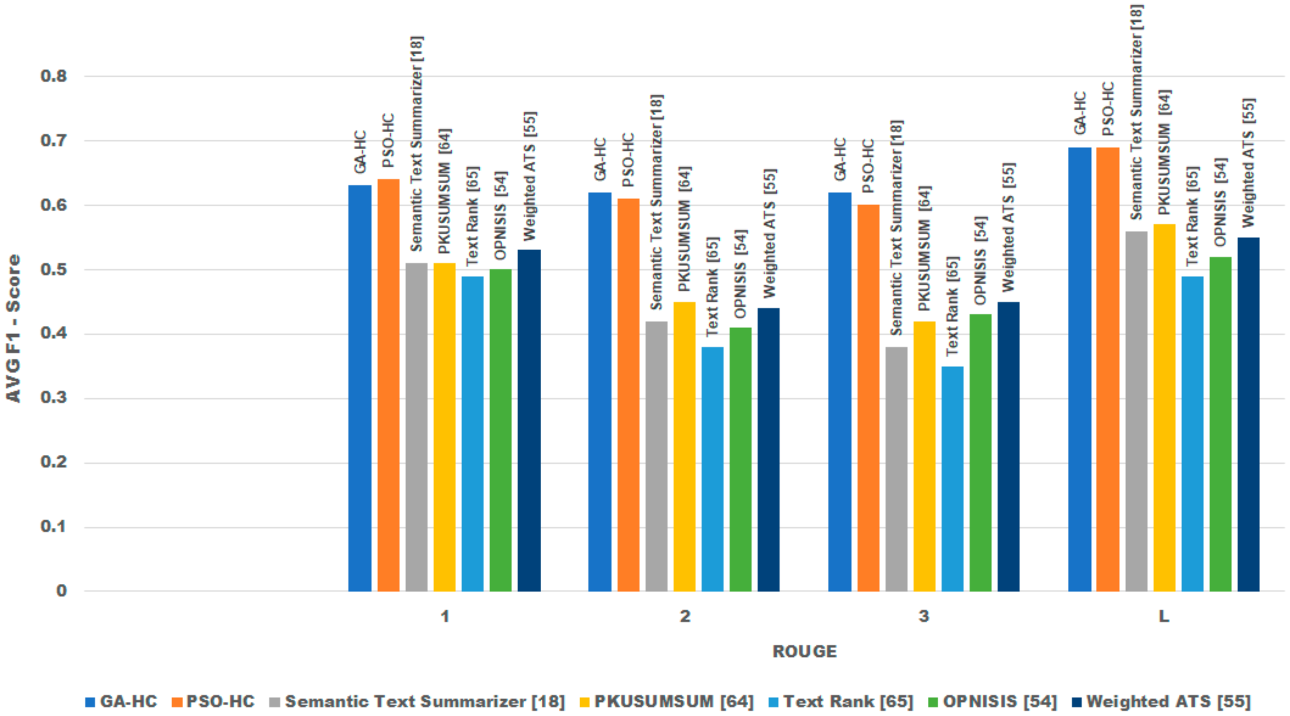


(**a**)

Figure 5. *Cont.*

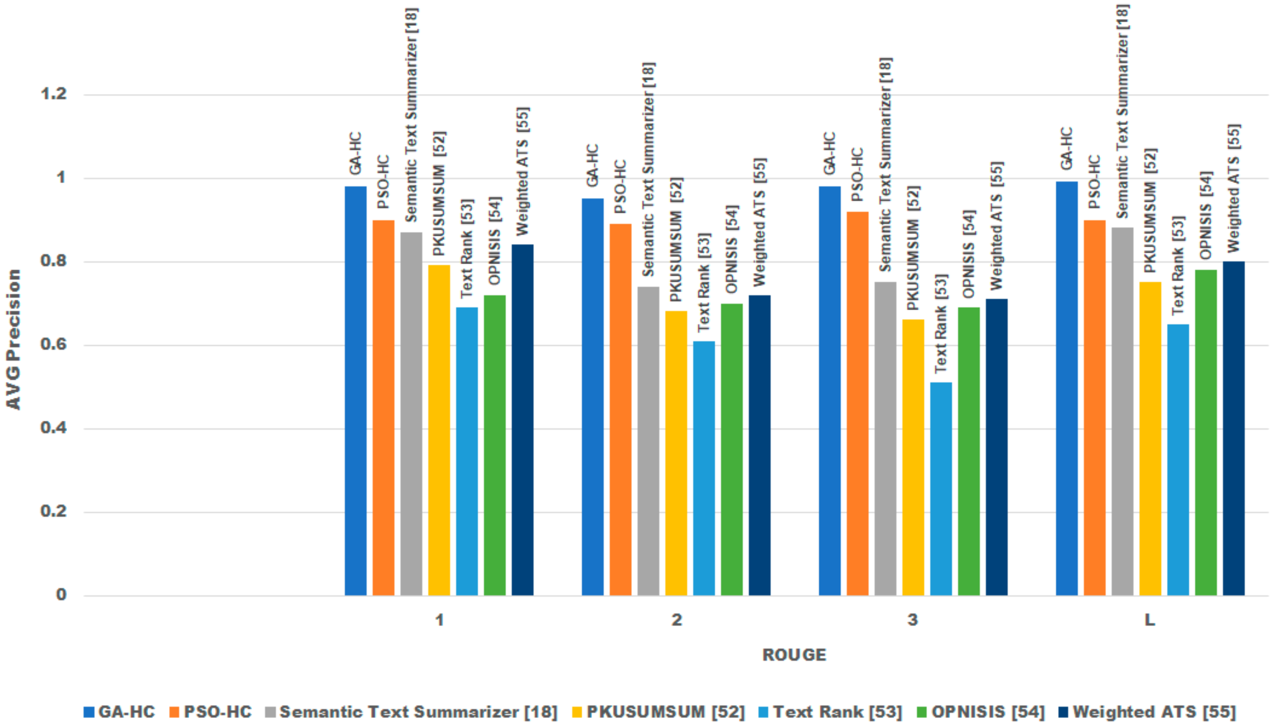


(**b**)

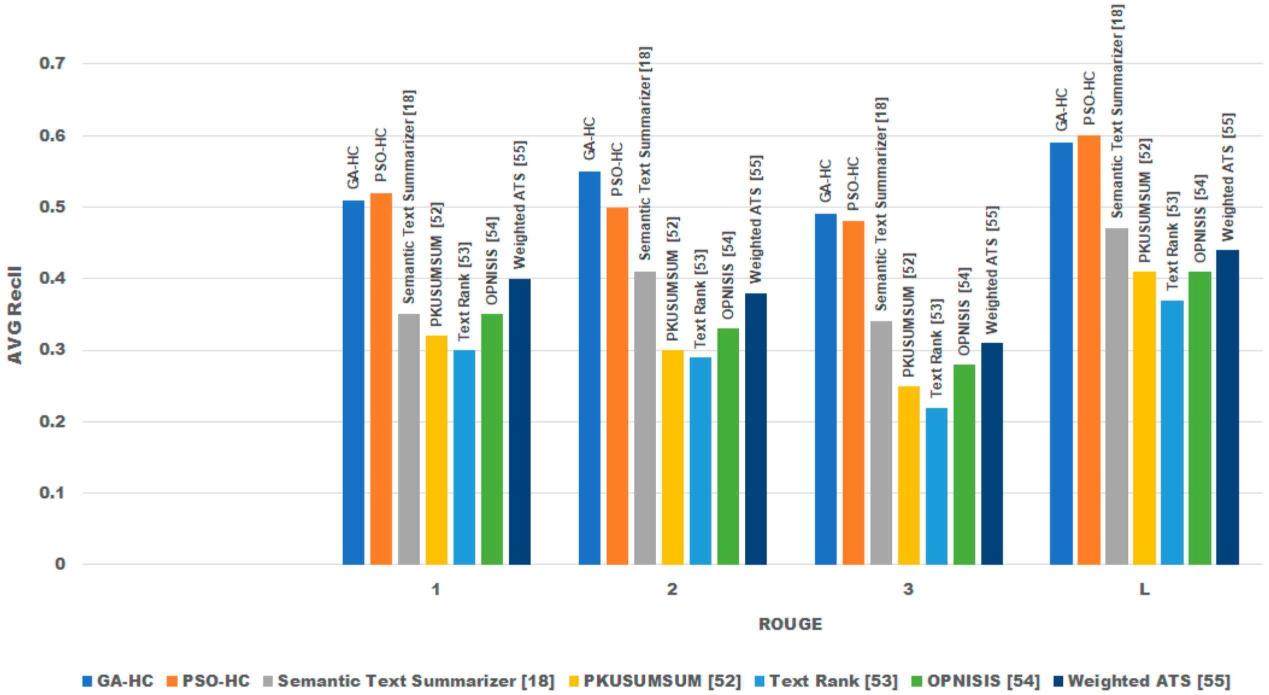


(**c**)

Figure 5. Comparison of various methods over DUC dataset for average values of (**a**) precision, (**b**) recall, and (**c**) F1-score.

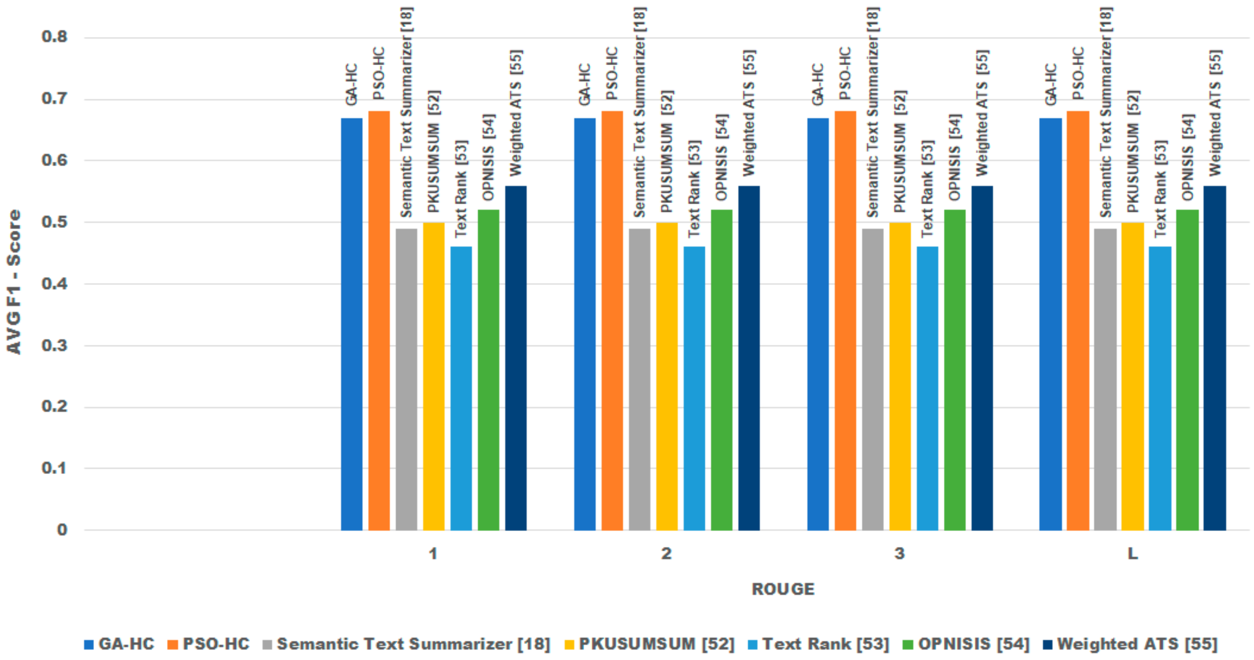


(**a**)



(**b**)

Figure 6. *Cont.*



(**c**)

Figure 6. Comparison of various methods over CNN/daily mail dataset for average values of (**a**) precision, (**b**) recall, and (**c**) F1-score.

Tables 1 and 2 showed the results of GA and PSO with three clustering approaches respectively over DUC dataset. Experimental results demonstrate the proposed models out perfume other text summarization models in terms of precision, recall, and F1-score.

Table 1. Genetic Algorithm (GA) with clustering techniques.

Genetic Algorithm (GA)				
Metric	**Rouge Type**	**Hierarchical Clustering**	**K-Means**	**Fuzzy C-Means**
Avg. PR	1	0.99	0.88	0.81
	2	0.99	0.80	0.67
	3	0.98	0.78	0.64
	L	0.99	0.89	0.81
Avg. Recall	1	0.48	0.44	0.42
	2	0.48	0.38	0.34
	3	0.47	0.37	0.32
	L	0.54	0.49	0.47
Avg. F1-Score	1	0.63	0.56	0.54
	2	0.62	0.51	0.43
	3	0.62	0.49	0.41
	L	0.69	0.62	0.58

Table 2. Particle Swarm Optimization (PSO) with clustering techniques.

Particle Swarm Optimization (PSO)				
Metric	**Rouge Type**	**Hierarchical Clustering**	**K-Means**	**Fuzzy C-Means**
Avg. PR	1	0.96	0.75	0.79
	2	0.94	0.57	0.64
	3	0.93	0.52	0.61
	L	0.96	0.75	0.78
Avg. Recall	1	0.50	0.36	0.39
	2	0.47	0.25	0.30
	3	0.46	0.23	0.28
	L	0.55	0.40	0.43
Avg. F1-Score	1	0.64	0.47	0.51
	2	0.61	0.34	0.40
	3	0.60	0.31	0.37
	L	0.69	0.51	0.55

5. Conclusions

In this paper, two variants of automatic text summarization model are presented. The proposed approach employed distributional semantics of the words present in the sentences of the text and used hierarchical clustering technique for grouping similar sentences. GA-HC applied GA for optimizing the results of extracted features while PSO-HA used PSO for optimizing the results of extracted features. Finally, the top ranked sentences are selected on the basis of certain threshold and combined to make a summary. The position of sentences is kept same as they appeared in the original text.

Our works can be concluded as: a. Applying underlying meaning of words and semantics as feature in text summarization to generate improved and better summaries. b. Hierarchal clustering technique can produce better results and c. Evolutionary techniques used for optimizing the features scores can be used to produce better summaries.

In future work, human evaluation will be considered which can further strengthen model performance. Moreover, multiple aspects including readability, correctness, completeness, and compactness of documents can be considered to improve the quality of summary. Moreover, the deep learning models will be considered for the data extraction and optimized using metaheuristic techniques [56–62].

Author Contributions: Conceptualization, M.M., S.L., and M.A.K.; methodology, M.M., M.H., and S.L.; software, M.M. and M.H.; validation, M.A.K., U.T., and S.K.; formal analysis, U.T. and S.K.; investigation, U.T. and M.A.K.; resources, H.-S.Y. and S.L.; data curation, M.A.K. and H.-S.Y.; writing—original draft preparation, M.M., S.L., and M.H.; writing—review and editing, H.-S.Y., M.A.K. and J.-I.C.; visualization, J.-I.C.; supervision, M.A.K. and M.H.; project administration, S.L. and J.-I.C.; funding acquisition, H.-S.Y. and J.-I.C. All authors have read and agreed to the published version of the manuscript.

Funding: This study was partially supported by Ewha Womans University.

Informed Consent Statement: Not applicable.

Data Availability Statement: The CASIA b dataset is utilized for the experimental process. This dataset is publically available for research purpose. Here is the link of dataset: Center for Biometrics and Security Research (http://www.cbsr.ia.ac.cn/english/index.asp, accessed on 5 November 2021).

Conflicts of Interest: The authors declare no conflict of interest in this work.

References

1. Chen, X.; Ke, L.; Lu, Z.; Su, H.; Wang, H. A novel hybrid model for cantonese rumor detection on twitter. *Appl. Sci.* **2020**, *10*, 7093. [CrossRef]
2. Hernandez, J.; Marin-Castro, H.M.; Morales-Sandoval, A.M. A semantic focused web crawler based on a knowledge representation schema. *Appl. Sci.* **2020**, *10*, 3837. [CrossRef]
3. Luhn, H.P. The automatic creation of literature abstracts. *IBM J. Res. Dev.* **1958**, *2*, 159–165. [CrossRef]
4. Narayan, S.; Papasarantopoulos, N.; Cohen, S.B.; Lapata, M. Neural extractive summarization with side information. *arXiv* **2017**, arXiv:1704,04530.
5. Yousefi-Azar, M.; Hamey, L. Text summarization using unsupervised deep learning. *Expert Syst. Appl.* **2017**, *68*, 93–105. [CrossRef]
6. Li, W.; Li, D.; Yin, H.; Zhang, L.; Zhu, Z.; Liu, P. Lexicon-enhanced attention network based on text representation for sentiment classification. *Appl. Sci.* **2019**, *9*, 3717. [CrossRef]
7. Martinčić-Ipšić, S.; Miličić, T.; Todorovski, L. The Influence of feature representation of text on the performance of document classification. *Appl. Sci.* **2019**, *9*, 743. [CrossRef]
8. Joshi, A.; Fidalgo, E.; Alegre, E.; Fernández-Robles, L. An unsupervised framework for extractive text summa-rization based on deep auto-encoders. *Expert Syst. Appl.* **2019**, *129*, 200–215. [CrossRef]
9. Vázquez, E.; García-Hernández, R.A.; Ledeneva, Y. Sentence features relevance for extractive text summarization using genetic algorithms. *J. Intell. Fuzzy Syst.* **2018**, *35*, 353–365. [CrossRef]
10. Wang, Q.; Liu, P.; Zhu, Z.; Yin, H.; Zhang, Q.; Zhang, L. A text abstraction summary model based on BERT word embedding and reinforcement learning. *Appl. Sci.* **2019**, *9*, 4701. [CrossRef]
11. Han, X.W.; Zheng, H.T.; Chen, J.Y.; Zhao, C.Z. Diverse decoding for abstractive document summariza-tion. *Appl. Sci.* **2019**, *9*, 386. [CrossRef]
12. Rouane, O.; Belhadef, H.; Bouakkaz, M. Combine clustering and frequent itemsets mining to enhance biomedical text summarization. *Expert Syst. Appl.* **2019**, *135*, 362–373. [CrossRef]
13. Du, Y.; Huo, H. News text summarization based on multi-feature and fuzzy logic. *IEEE Access* **2020**, *8*, 140261–140272. [CrossRef]
14. Leiva, L.A. Responsive text summarization. *Inf. Process. Lett.* **2018**, *130*, 52–57. [CrossRef]
15. Fang, C.; Mu, D.; Deng, Z.; Wu, Z. Word-sentence co-ranking for automatic extractive text summarization. *Expert Syst. Appl.* **2017**, *72*, 189–195. [CrossRef]
16. Singh, P.; Chhikara, P.; Singh, J. An ensemble approach for extractive text summarization. In Proceedings of the 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), Vellore, India, 24–25 February 2020; pp. 1–7.
17. Jindal, S.G.; Kaur, A. Automatic keyword and sentence-based text summarization for software bug reports. *IEEE Access* **2020**, *8*, 65352–65370. [CrossRef]
18. Mohd, M.; Jan, R.; Shah, M. Text document summarization using word embedding. *Expert Syst. Appl.* **2020**, *143*, 112958. [CrossRef]
19. Qaroush, A.; Abu Farha, I.; Ghanem, W.; Washaha, M.; Maali, E. An efficient single document Arabic text summarization using a combination of statistical and semantic features. *J. King Saud Univ.-Comput. Inf. Sci.* **2019**. [CrossRef]
20. Rajangam, M.; Annamalai, C. Extractive document summarization using an adaptive, knowledge based cognitive model. *Cogn. Syst. Res.* **2019**, *56*, 56–71. [CrossRef]
21. Sanchez-Gomez, J.M.; Vega-Rodríguez, M.A.; Pérez, C.J. Extractive multi-document text summarization using a multi-objective artificial bee colony optimization approach. *Knowl.-Based Syst.* **2018**, *159*, 1–8. [CrossRef]
22. Chen, J.; Zhuge, H. Extractive summarization of documents with images based on multi-modal RNN. *Futur. Gener. Comput. Syst.* **2019**, *99*, 186–196. [CrossRef]
23. Priya, V.; Umamaheswari, K. Enhanced continuous and discrete multi objective particle swarm optimization for text summarization. *Clust. Comput.* **2019**, *22*, 229–240. [CrossRef]
24. Verma, P.; Om, H. MCRMR: Maximum coverage and relevancy with minimal redundancy based multi-document summari-zation. *Expert Syst. Appl.* **2019**, *120*, 43–56. [CrossRef]
25. Lamsiyah, S.; El Mahdaouy, A.; Espinasse, B.; Ouatik, S.E.A. An unsupervised method for extractive multi-document summarization based on centroid approach and sentence embeddings. *Expert Syst. Appl.* **2021**, *167*, 114152. [CrossRef]
26. Rekabdar, B.; Mousas, C.; Gupta, B. Generative adversarial network with policy gradient for text summarization. In Proceedings of the 2019 IEEE 13th International Conference on Semantic Computing (ICSC), Newport Beach, CA, USA, 30 January–1 February 2019; pp. 204–207.
27. Goularte, F.B.; Nassar, S.M.; Fileto, R.; Saggion, H. A text summarization method based on fuzzy rules and applicable to automated assessment. *Expert Syst. Appl.* **2019**, *115*, 264–275. [CrossRef]
28. Sonawane, S.; Kulkarni, P.; Deshpande, C.; Athawale, B. Extractive summarization using semigraph (ESSg). *Evol. Syst.* **2018**, *10*, 409–424. [CrossRef]
29. Rautray, R.; Balabantaray, R.C. Cat swarm optimization based evolutionary framework for multi document summarization. *Phys. A Stat. Mech. Its Appl.* **2017**, *477*, 174–186. [CrossRef]
30. Patel, D.; Shah, S.; Chhinkaniwala, H. Fuzzy logic based multi document summarization with improved sentence scoring and redundancy removal technique. *Expert Syst. Appl.* **2019**, *134*, 167–177. [CrossRef]

31. Li, H.; Zhu, J.; Ma, C.; Zhang, J.; Zong, C. Read, watch, listen, and summarize: Multi-modal summarization for asynchronous text, image, audio and video. *IEEE Trans. Knowl. Data Eng.* **2018**, *31*, 996–1009. [CrossRef]
32. Vetriselvi, T.; Gopalan, N.P. An improved key term weightage algorithm for text summarization using local context information and fuzzy graph sentence score. *J. Ambient. Intell. Humaniz. Comput.* **2021**, *12*, 4609–4618. [CrossRef]
33. Al-Sabahi, K.; Zuping, Z.; Nadher, M. A hierarchical structured self-attentive model for extractive document summarization (HSSAS). *IEEE Access* **2018**, *6*, 24205–24212. [CrossRef]
34. Diao, Y.; Lin, H.; Yang, L.; Fan, X.; Chu, Y.; Wu, D.; Zhang, D.; Xu, K. CRHASum: Extractive text summarization with contextualized-representation hierarchical-attention summarization network. *Neural Comput. Appl.* **2020**, *32*, 11491–11503. [CrossRef]
35. Alguliyev, R.M.; Aliguliyev, R.M.; Isazade, N.R.; Abdi, A.; Idris, N. COSUM: Text summarization based on clustering and optimization. *Expert Syst.* **2019**, *36*, e12340. [CrossRef]
36. Mohamed, M.; Oussalah, M. SRL-ESA-TextSum: A text summarization approach based on semantic role labeling and explicit semantic analysis. *Inf. Process. Manag.* **2019**, *56*, 1356–1372. [CrossRef]
37. Manning, C.D.; Surdeanu, M.; Bauer, J.; Finkel, J.R.; Bethard, S.; McClosky, D. The Stanford CoreNLP natural language pro-cessing toolkit. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Baltimore, Maryland, USA, 23–24 June 2014; pp. 55–60.
38. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
39. Canales, L.; Strapparava, C.; Boldrini, E.; Martinez-Barco, P. Intensional learning to efficiently build up automatically annotated emotion corpora. *IEEE Trans. Affect. Comput.* **2017**, *11*, 335–347. [CrossRef]
40. Schütze, H. Automatic word sense discrimination. *Comput. Linguist.* **1998**, *24*, 97–123.
41. Barzilay, R.; Lapata, M. Modeling local coherence: An entity-based approach. *Comput. Linguist.* **2008**, *34*, 1–34. [CrossRef]
42. Edmundson, H.P.; Wyllys, R.E. Automatic abstracting and indexing—Survey and recommendations. *Commun. ACM* **1961**, *4*, 226–234. [CrossRef]
43. McCreadie, R.; Macdonald, C.; Ounis, I. Automatic ground truth expansion for timeline evaluation. In Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, Ann Arbor, MI, USA, 8–12 July 2018; pp. 685–694.
44. Kulkarni, A.R.; Apte, M. An Automatic Text Summarization Using Feature Terms for Relevance Measure. *IOSR J. Comput. Eng.* **2002**, *9*, 62–66. [CrossRef]
45. Ferreira, R.; Cabral, L.; Lins, R.D.; e Silva, G.P.; Freitas, F.; Cavalcanti, G.D.; Lima, R.; Simske, S.J.; Favaro, L. Assessing sentence scoring techniques for extractive text summarization. *Expert Syst. Appl.* **2013**, *40*, 5755–5764. [CrossRef]
46. Salton, G.; Buckley, C. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manag.* **1988**, *24*, 513–523. [CrossRef]
47. Goldberg, D.E.; Holland, J.H. Genetic algorithms and machine learning. *Mach. Learn.* **1988**, *3*, 95–99. [CrossRef]
48. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95-International Conference on Neural Net-works, Perth, WA, Australia, 27 November–1 December 1995; pp. 1942–1948.
49. Lin, C.-Y. Rouge: A package for automatic evaluation of summaries. In Proceedings of the Workshop on Text Summarization of ACL, Barcelona, Spain, 25–26 July 2004; pp. 74–81.
50. Hartigan, J.A.; Wong, M.A. Algorithm AS 136: A K-means clustering algorithm. *J. R. Stat. Soc. Ser. C (Appl. Stat.)* **1979**, *28*, 100. [CrossRef]
51. Bezdek, J.C.; Ehrlich, R.; Full, W. FCM: The fuzzy c-means clustering algorithm. *Comput. Geosci.* **1984**, *10*, 191–203. [CrossRef]
52. Zhang, J.; Wang, T.; Wan, X. PKUSUMSUM: A Java platform for multilingual document summarization. In Proceedings of the Coling 2016, the 26th International Conference on Computational Linguistics: System Demonstrations, Osaka, Japan, 11–16 December 2016; pp. 287–291.
53. Mihalcea, R.; Tarau, P. Textrank: Bringing order into text. In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain, 25–26 July 2004; pp. 404–411.
54. Ganesan, K.; Zhai, C.; Han, J. Opinosis: A Graph-Based Approach to Abstractive Summarization of Highly Redundant Opinions. 2010. Available online: https://www.ideals.illinois.edu/handle/2142/16949 (accessed on 5 November 2021).
55. Rani, R.; Lobiyal, D.K. A weighted word embedding based approach for extractive text summarization. *Expert Syst. Appl.* **2021**, *186*, 115867. [CrossRef]
56. Khan, M.A.; Muhammad, K.; Sharif, M.; Akram, T.; Kadry, S. Intelligent fusion-assisted skin lesion localization and classification for smart healthcare. *Neural Comput. Appl.* **2021**, 1–16. [CrossRef]
57. Khan, M.A.; Sharif, M.; Akram, T.; Kadry, S.; Hsu, C. A two-stream deep neural network-based intelligent system for complex skin cancer types classification. *Int. J. Intell. Syst.* **2021**. [CrossRef]
58. Nawaz, M.; Nazir, T.; Masood, M.; Mehmood, A.; Mahum, R.; Khan, M.A.; Kadry, S.; Thinnukool, O. Analysis of brain MRI images using improved cornernet approach. *Diagnostics* **2021**, *11*, 1856. [CrossRef]
59. Wang, S.-H.; Khan, M.A.; Govindaraj, V.; Fernandes, S.L.; Zhu, Z.; Zhang, Y.-D. Deep rank-based average pooling network for COVID-19 recognition. *Comput. Mater. Contin.* **2022**, *70*, 2797–2813. [CrossRef]
60. Manic, K.S.; Biju, R.; Patel, W.; Khan, M.A.; Raja, N.S.M.; Uma, S. Extraction and evaluation of corpus callosum from 2D brain MRI slice: A study with cuckoo search algorithm. *Comput. Math. Methods Med.* **2021**, *2021*, 1–15. [CrossRef] [PubMed]

61. Khan, M.A.; Zhang, Y.-D.; Alhusseni, M.; Kadry, S.; Wang, S.-H.; Saba, T.; Iqbal, T. A fused heterogeneous deep neural network and robust feature selection framework for human actions recognition. *Arab. J. Sci. Eng.* **2021**, 1–16. [CrossRef]
62. Khan, M.A.; Muhammad, K.; Sharif, M.; Akram, T.; de Albuquerque, V.H.C. Multi-class skin lesion detection and classification via teledermatology. *IEEE J. Biomed. Health Inform.* **2021**. [CrossRef] [PubMed]