International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 6
DOI: https://doi.org/10.17762/ijritcc.v11i6.7246
***Article Received**: 25 March 2023 **Revised**: 30 April 2023 **Accepted**: 12 June 2023*

Hybrid Optimization Based Hindi Document Summarization Using Deep Learning Technique

Sumalatha Bandari[1], Vishnu Vardhan Bulusu[2]
[1]Department of Computer Science and Engineering
Jawaharlal Nehru Technological University
Hyderabad, India
sumaabandari@gmail.com
[2]Department of Computer Science and Engineering
JNTUH College of Engineering
Manthani, India
mailvishnu@jntuh.ac.in

Abstract— The proliferation of textual information today is a result of the internet's recent development, which is widely accessible to anybody, at any time. Generally speaking, several Natural Language Processing (NLP) techniques can be used to analyze the textual information that is offered on the basis of text documents. In recent years, various text summarization techniques have been implemented in English text documents but a little amount of work is carried out in Hindi text documents summarization. In this research investigation, the Coot Remora Optimization (CRO) technique based on Deep Recurrent Neural Network (DRNN) is used to summarize Hindi documents. Here, the CRO algorithm is used to train the DRNN, which is used to compute the sentence scores.The highest scored sentences are going to included in the summary. When compared to recent optimization algorithmic techniques, such as MCRMR-SSO, Graph-based_PSO, Genetic Algorithms (GA), and Political Elephant Herding Optimization (PEHO) based Deep Long Short Term Memory (DLSTM) algorithm, the developed method is shown to be superior. Additionally, three evaluation metrics such as precision, recall, f-measure are used to analyze the performance of the CRO based DRNN technique and obtained high performance.

Keywords- Hindi document summarization, Political Elephant Herding Optimization, Coot Remora Optimization.

I. INTRODUCTION

More than 250 million people in India use Hindi as their first language, making it the country's official language [1]. Hindi language contains a vast amount of information, thus it's essential to quickly understand the key passages and plan for time-sensitive Hindi text document circumstances without skipping any significant details [2]. Word documents can be categorized depending on the texts provided to assist you recover all of its information. The vast majority of activities in document text management are based on the English corpus, but to make it simple to access Hindi on a web page, Unicode standards for Indic languages must be introduced. As a result, Hindi content was substantially more readily available online. For NLP tasks including word identification, stemming, and summarization, researchers have mostly focused on Hindi text [3]. Text summary is the process of reducing lengthy texts to more manageable sizes while preserving information [4]. Text summarizing has become an essential tool for reading, comprehending, and analyzing text materials in daily life, such as news headlines, introduction summaries, book reviews, and so forth [5]. Hindi-based documents are frequently utilized in a variety of contexts where text summaries aids in reader comprehension and provides an overview of the document.

In document processing and information retrieval systems, automatic summarization is essential. Automated document summarization seeks to produce succinct summaries from extensive source material. A good summary should encompass the main ideas of the original article or collection of papers while being coherent, non-redundant, and grammatically accurate [6]. The exponential growth of textual data on different computing devices makes automatic text processing crucial, and it has the power to solve a variety of information overload issues [7]. In order to create a summary while retaining the core ideas, automatic text summarizing eliminates redundant or extraneous language from the source text [8]. The demand for a summarizing system is growing as a result of the difficulties in decision-making brought on by a web page's overflow of information. The summary approach offers the entire material in a condensed version, enabling users to make decisions quickly and simply [9].

Automated summarizing of text is divided into two distinct forms based on the type of data and text: extractive text automated summarization and abstractive text automatic summarization. The extractive summary has high-ranking original documents with feature values in sentences, phrases, paragraphs, and other areas, whereas the abstractive summary provides a summary of text analysis in linguistic data [5]. The text's linguistic and statistical components serve as the selection criteria for extracting summaries. Language elements are employed in abstractive summaries to understand material and formulate new sentences [10]. Single-document and multi-document summarizing are two different categories for the process of text summarization. Single-document summarization requires the summary of a single document, whereas multi-document summarizing requires the summary of multiple documents [4]. A long text document is condensed into a shorter one in a single document summary [11].

Before using these procedures, it is crucial to first identify the relevant papers from the collection of input documents [9]. The majority of research solely focuses on widely spoken European languages. Since there isn't as much information available in languages other than English, Indian languages haven't been studied as extensively. The situation is currently shifting, and a variety of materials are now available in numerous additional languages. The need for text summarizing methods that are compatible with Indian languages seems to be growing [2]. Instead of reading the entire page, extractive summarizing techniques could be used to automatically construct summaries of Hindi literature. Currently, the system would produce summaries for individual text documents in Hindi. For systems that make conclusions on the analysis of lengthy texts, a summary of the Hindi text is crucial to the analysis process. In addition, there are numerous applications for Hindi text summarizing in systems, including knowledge representation and text analysis [12].

The identification of significant sentences and the identification of groups of Hindi phrases are essential for extractive summarization since they improve accuracy [10]. Summarization performed utilizing hybrid approaches aids in the extraction of statistical information from the documents with important extraction processes, such as sentence length, location, frequency, and so forth [2]. A lexical tool for the Hindi language called Hindi SentiWordNet comprises words together with the accompanying sentiment scores. It is useful for a variety of NLP applications, including sentiment analysis, document classification, and document summarization [5]. Hindi SentiWordNet can be used to determine the sentiment of each sentence in a Hindi document when summarizing it. Positive or negative sentences with high sentiment scores can be categorized as key sentences and added to the summary. The sentiment scores of the sentences that make up the summary can also be used to calculate the summary's sentiment score.

The primary goal of the experimental research is to obtain the summarized text of Hindi documents using CRO based DRNN model using hybrid optimization techniques. The performance of the CRO-based DRNN is compared to that of other conventional methods such as the MCRMR-SSO, Graph-based PSO, GA, and PEHO-based DLSTM in order to assess the model's efficacy. The effectiveness of the CRO-based DRNN approach is also assessed using other assessment criteria, including precision, recall and f-measure.

The research paper is divided into the following sections: section 2, which summarizes the literature; section 3, which illustrates the proposed model; section 4, which describes the results of the experiment and performs a superiority analysis; and section 5, which is the research paper's conclusion.

II. LITERATURE REVIEW

Gupta and Garg [4] presented the topic of text summarization of Hindi documents using rule-based approaches. This system describes how the Hindi text summarization is carried out using linguistic features. To get the summarized text dead words and phrases are removed from the original document. A rule-based approach that used linguistic features such as sentence length, sentence position, and sentence structure to identify important sentences in a Hindi document. A rule-based approach that used syntactic features to identify important sentences in a Hindi document. The limitations of this system are semantic rules of Hindi text were not used to generate a summary and limited corpus size.

Vimal Kumar and Yadav [2] present an improvised extractive approach to Hindi text summarization. The authors discussed the importance of text summarization in the context of the growing amount of online and offline text data. Three steps were taken to implement this strategy: ranking the sentences, examining the sentiment of the sentences, and extracting the sentences. By applying the compression ratio, the most important and applicable sentence summary was extracted from the original text. Despite this strategy was highly effective in providing a relevant summary, it failed to apply approaches to neglect sentences that do not provide a lot of information.

Gulati and Sawarkar [13] proposes a new technique for summarizing multiple documents in the Hindi language. This system used three main processes of the summarization system are preprocessing, extracting feature words, and using fuzzy logic to order the phrase according to the optimum feature weights. Online Hindi newspapers news such as stories on politics and sports were used as input for the system. The limitation of this system was the neural network algorithm for Hindi was not used.

Kumar, K. V et. al. [9] proposed a novel approach to automatically summarize Hindi text documents using graph-based methods. To analyze the importance of sentence and to find the relation between the sentences, the sentences are ranked according to how frequently they occur in the input document. The ranks are then utilized to determine which sentences are the most crucial to the document. Semantic similarity was used to find the relevance of one sentence to another sentence in the document. Semantic analysis of the sentences and ranks are used to merge the identified relevant sentences. The system can be improved by employing an abstractive technique to find the key sentences and semantic analysis to find the connections between them.

Pareek G. [1] presents a novel approach for text summarization in Hindi language using Genetic Algorithm (GA). The Genetic Algorithm, which is a well-known optimization technique inspired by the process of natural selection. The steps include in this method are text preprocessing, sentence scoring, chromosome encoding, fitness function evaluation, and crossover and mutation operations. This system's dataset consisted of Hindi news articles. Summarization quality and computational efficiency of this system was good. This system focuses on the Hindi language, and it is not clear how well the genetic algorithm approach was generalized to other languages.

Dalal V. and Malik L. [14] proposes a novel approach for automatic text summarization in Hindi using semantic graph-based techniques and particle swarm optimization (PSO). One of the most potent bio-inspired algorithms for finding the best solution is PSO. The document's Subject-Object-Verb (SOV) triples are taken out. The document's semantic graph is built using these triples. The PSO algorithm is used to train a classifier, which is then used to produce a semantic sub-graph and acquire a document summary. The limitations of this system are the approach relies heavily on the quality and completeness of the Hindi WordNet database used to construct the semantic graph and the approach may not be suitable for very short or very long documents.

Pradeepika Verma and Hari Om [15] proposed MCRMR: Maximum coverage and relevancy with minimal redundancy based multi-document summarization. This system was extraction-based method for multi-document summarization that considers coverage, non-redundancy, and relevance—three key aspects of an effective summary. A single document is created from numerous documents using the coverage and non-redundancy characteristics. The weighted mixture of word embedding and Google-based similarity algorithms were used to investigate these aspects. The text summarization task is described as an optimization problem, where multiple text features with their optimized weights are employed to score the sentences in order to locate the relevant sentences. This is done to accommodate the relevancy feature in the system generated summaries. The meta-heuristic technique known as Shark Smell Optimization (SSO) was used for the features weight optimization. Six benchmark datasets (DUC04, DUC06, DUC07, TAC08, TAC11 and MultiLing13) were used for the experiments. The limitation of this system was neural network-based models were not used.

Sumalatha B. and Vishnu Vardhan B. [16] proposed Feature extraction based Deep Long Short-Term Memory for Hindi document summarization using Political Elephant Herding Optimization (PEHO). This is relatively new optimization algorithm that is inspired by the herding behavior of elephants in political contexts. While there is not yet a significant body of research on the use of PEHO specifically for feature extraction in Hindi document summarization, there has been some research on using deep learning techniques such as Long Short-Term Memory (LSTM) networks for this task. LSTM networks are a type of recurrent neural network (RNN) that are designed to overcome the vanishing gradient problem and better capture long-term dependencies in sequential data. In the context of text summarization, LSTM networks can be used to process the input document and generate a summary by learning to identify important phrases and sentences. Feature extraction techniques can be used to identify and extract the most relevant information from the input document to be used as input to the LSTM network. These techniques can include methods such as TF-IDF, Thematic and word length. Once these features have been extracted, they can be fed into the LSTM network as input. To apply the PEHO optimization algorithm to this process, the network's weights can be optimized using the PEHO algorithm. This can involve adjusting the weights of the LSTM network to better capture the relationships between the extracted features and the target summary. The optimization process can be repeated iteratively until the optimal set of weights is found. The combination of feature extraction techniques, LSTM networks, and optimization algorithms such as PEHO can be used to improve the accuracy and effectiveness of Hindi document summarization.

Rupal Bhargava et al. [17] presented an approach for abstractive text summarizing using a Generative Adversarial Network (GAN) to process multilingual text summarization. Improvements were made in terms of the size of the dataset and hyper-parameter tweaking. Also, this strategy delivered consistent high performance. This approach's primary drawback is that it was only applicable to huge datasets.

III. PROPOSED MODEL

The CRO based DRNN technique used for Hindi text document summarization is developed using the following process, such as text acquisition, tokenization, text feature extraction, score generation and Summary generation. The flow diagram of CRO based DRNN technique for Hindi document summarization is briefly illustrated in the Fig. 1.

DOI: https://doi.org/10.17762/ijritcc.v11i6.7246
***Article Received**: 25 March 2023 **Revised**: 30 April 2023 **Accepted**: 12 June 2023*

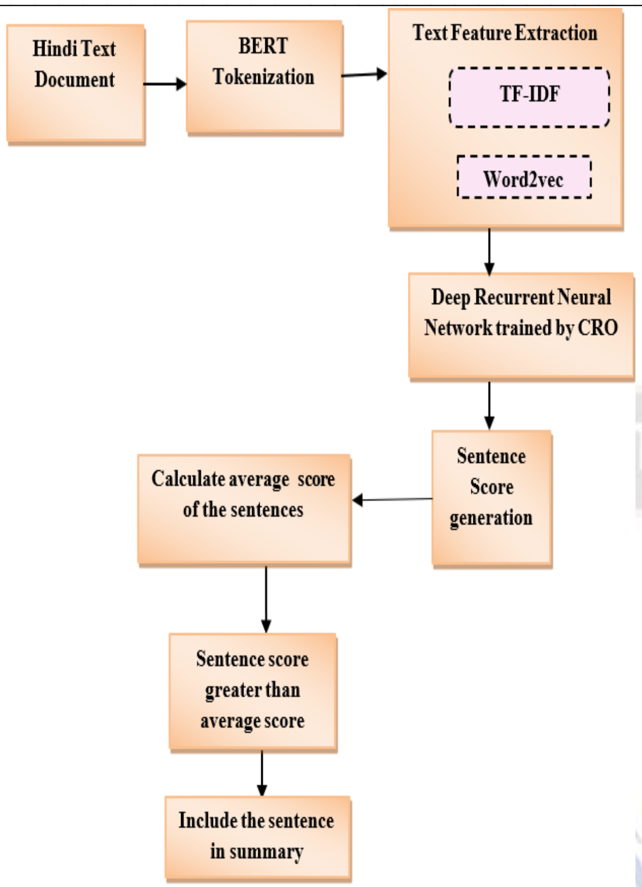
Hindi Text Document
BERT Tokenization
Text Feature Extraction
TF-IDF
Word2vec
Deep Recurrent Neural Network trained by CRO
Sentence Score generation
Calculate average score of the sentences
Sentence score greater than average score
Include the sentence in summary

Figure 1. Flow diagram of CRO based DRNN technique for Hindi document summarization

The Hindi text documents are taken as input for the proposed method. These documents are applied to BERT tokenizer to split the input documents into tokens. BERT (Bidirectional Encoder Representations from Transformers) is a state-of-the-art deep learning model that has been used successfully for various NLP tasks, including text summarization [18]. BERT uses a transformer-based architecture that enables it to capture complex relationships between words and generate highly accurate representations of input text. When using BERT for Hindi document summarization, the first step is to tokenize the input text document using a tokenizer specifically designed for the Hindi language. There are several BERT-based tokenizers available for Hindi, including Indic-BERT tokenizer, Hindi-BERT tokenizer, and Multilingual-BERT tokenizer. These tokenizers segment the input text into sub words and generate corresponding token embeddings that can be used as input to the BERT model. Once the input text has been tokenized, the BERT model is used to generate contextualized representations of each token. These representations capture the meaning and context of each token in the input text, and are used as input to the summarization model. BERT tokenization can be used effectively in Hindi document summarization by segmenting the input text into sub words and generating corresponding token embeddings.

Then, various features Term Frequency–Inverse Document Frequency (TF-IDF) [19], and word2vec [20], are extracted from the tokens. TF-IDF is a widely used technique in natural language processing for extracting keywords and ranking their importance in a given document. The technique assigns a score to each word or term based on its frequency in the document and its frequency across all the documents in the corpus. Term frequency: The frequency of each term in the document is calculated, which gives the raw count of the number of times each term appears in the document. Inverse document frequency: The inverse document frequency is calculated for each term, which reflects the importance of the term in the corpus. The inverse document frequency is calculated as log(N/df), where N is the total number of documents in the corpus, and df is the number of documents in which the term appears. The TF-IDF score is then calculated for each term, which is the product of the term frequency and the inverse document frequency. The sentences or phrases with the highest TF-IDF scores are selected to generate the summary. TF-IDF can be effective in summarizing Hindi text documents, as it allows for the identification of the most important terms and phrases in the document. However, it is noted that TF-IDF does not take into account the semantic relationships between terms, which may limit its effectiveness in capturing the full meaning of the text. Combining TF-IDF with other techniques, such as semantic graph-based approaches or machine learning-based approaches, may improve the quality of the summary.

Calculate the cosine similarity between each pair of sentences in the document using the TF-IDF matrix. The cosine similarity score represents the similarity between two sentences based on the words they share. Rank the sentences based on their cosine similarity scores. The sentences with the highest scores are the most similar to each other and are likely to contain similar information. Select the top-ranked sentences to generate the summary. The number of sentences selected depends on the desired length of the summary.

Word2vec is a popular technique for natural language processing that is used to represent words as dense vectors in a high-dimensional space. The technique uses a neural network to learn word embeddings, which capture the semantic relationships between words. Word2vec can be used in Hindi text document summarization to capture the meaning of the text and identify the most important sentences or phrases. The Word2Vec algorithm consists of two different models: Continuous Bag-of-Words (CBOW) and Skip-Gram. Both of these models are trained on a large corpus of text to learn word embeddings. In CBOW, the model predicts the current word based on the context words surrounding it. The context words are averaged and used to predict the target word. This model is useful for smaller datasets and when the context words are sufficient to predict the target word accurately. In Skip-Gram, the model predicts the context words surrounding the current word. The target word is used to predict the context words. This model is useful for larger datasets and when the target word has many possible context words. In CBOW, the input to the model is a bag of context words and the output is the predicted target word. In Skip-Gram, the input is the target word and the output is a bag of context words.

The Word2vec algorithm is used to learn word embeddings for each word in the text. The word embeddings are learned by training a neural network on a large corpus of text. The word embeddings are combined to generate sentence embeddings, which capture the meaning of each sentence in the document. The sentence embeddings are calculated as the average of the word embeddings in the sentence. The similarity between each sentence in the document and the summary is calculated based on the cosine similarity between the sentence embeddings and the summary embeddings. The sentences with the highest similarity scores are selected to generate the summary. In order to generate scores for the significance determination of the sentence, the extracted features from token are fed into DRNN [21] and is trained using CRO algorithm for adjusting the weights and learning parameters of DRNN. The devised CRO algorithm technique used to enhance the optimization process is the integration of ROA [22] and COOT [23].

COOT Optimization is a recently developed optimization algorithm that stands for "Cuckoo Optimization Technology". It is based on the behavior of cuckoo birds in nature, specifically their brood parasitism behavior, where they lay their eggs in the nests of other birds, allowing them to raise the cuckoo chicks instead. In COOT optimization, a population of candidate solutions is evolved over a number of iterations. Each candidate solution is evaluated based on a fitness function, and the best solutions are retained and used to generate new candidate solutions through a combination of mutation and crossover operations. The algorithm is inspired by the cuckoo's behavior of laying eggs in other birds' nests, as it allows for the discovery of new and potentially better solutions. In COOT optimization, this is accomplished by allowing some candidate solutions to be replaced by new and potentially better solutions, mimicking the cuckoo's brood parasitism behavior. The COOT algorithm has been applied to a variety of optimization problems, including feature selection, image segmentation, and classification. It has also been used in the field of text summarization. In the context of Hindi document summarization, COOT optimization is used to optimize the weights of the feature extraction and summarization algorithms. The algorithm can help to guide the search towards better feature subsets and summarization models by replacing weaker solutions with new and potentially better ones.

Remora optimization is applied to text summarization tasks by treating the summarization problem as an optimization problem. The objective is to generate a summary that captures the most important information from a longer text while minimizing the length of the summary. In this context, the text to be summarized can be represented as the host animal, and the candidate summaries can be represented as the remoras. The fitness function is then defined as a measure of the quality of the summary, such as its coherence, relevance, and informativeness. The remoras in the swarm can be initialized randomly, and then iteratively updated using a combination of local and global search strategies. For example, a remora might randomly select a sentence from the text and add it to its summary, or it might swap two sentences in its summary to improve its fitness score. At each iteration, the best summary found so far is retained and used to guide the search in subsequent iterations. Remora optimization has shown promise for text summarization tasks, particularly for multi-document summarization where multiple source texts are available. However, like other optimization algorithms, its effectiveness depends on the specific problem being solved and the choice of algorithm parameters. Other approaches such as graph-based methods and deep learning techniques are also commonly used for text summarization. By combining COOT optimization with Remora optimization for text document summarization CRO [24] algorithm is obtained which gives an improved performance for Hindi text document summarization.

CRO optimization is used to optimize the weights of the feature extraction and summarization algorithms. The optimized weights obtained from CRO are applied to Recurrent Neural Network (RNN). RNNs are a type of neural network that are well-suited for processing sequences of data, such as sentences in a document. They can be used for sentence classification, where the task is to classify whether a sentence should be included in the summary or not. RNNs have been used in various applications of text summarization and capture the temporal dependencies in the input text. The actual summary of the Hindi text documents are obtained from CRO based RNN. The nature inspired hybrid optimization algorithm CRO is used in the deep RNN based Hindi text document summarization to train the model. The algorithmic representation of the CRO based DRNN optimization is illustrated in the Table I.

TABLE I. ALGORITHM OF CRO BASED DEEP RECURRENT NEURAL NETWORK FOR HINDI DOCUMENTS SUMMARIZATION

```
Input: Hindi Text Documents
Output: Sentence scores, Summarized Text Document
For all Documents
initializations: n=number of words, m=number of sentences, p=0, Sentence Score=0

Begin
Step 1: i→ Tokens of the document
Step 2: j→TF/IDF values
Step 3: k→word embeddings
Step 4: p→ Sentence embeddings
Step 5: for (q=1 to m)
            $p=\sum^{n}_{i=1} k$
          Apply (j, p) to CRO optimizer
          Get optimized weights
          Apply optimized weights to train DRNN
          Update Sentence Score
Step 6: Calculate Average Score
Step 7: if (Sentence Score>= Average Score)
            Include the sentence in summary
      else
            Discard the sentence
End
```

Considering the input Hindi text document and divide into tokens. Calculate TF/IDF and word embedding vectors of these tokens. Summation of word embedding to get sentence embeddings. The words with higher value of TF/IDF are include in the summary. Sentence embeddings helps to get the semantic relation of the words in the summarized text. Apply TF/IDF values and sentence embedding to CRO to get optimized weights. Train DRNN with the optimized weights obtained from CRO. The output of DRNN provide sentence scores of each sentence in the document. Then after calculate average score of the sentences in the document by summation of individual sentence scores divided by number of sentences. If the sentence score is greater than or equal to average score then that sentence is included in the summary. Otherwise discard that sentence means the sentence does not include in the summary. The experimentation was done on the considered dataset and the results are shown in section 4.

IV. RESULTS

A. Dataset Description

The Hindi Text Short Summarization Corpus database, which was utilized in [25] for several evaluation metrics, provided the dataset for the analysis. Hindi Text Short Summarization Corpus is the first database devoted to the Hindi language, and it may also be used as a reference source for summarizing Hindi text. It includes articles organized with headlines taken from Hindi news websites, however it does not include the Hindi Text Short and Large Summarization Corpus. Instead, the articles have their own unique terminology, numerals, and punctuation.

1) Sample news article:

प्रधानमंत्री के साथ आर्मी और एयरफोर्स के चीफ भी मौजूद रह सकते हैं. एयरबेस पर पाकिस्तानी आतंकियों ने हफ्ते भर पहले हमला किया था, जिसमें सात जवान शहीद हुए थे. प्रधानमंत्री नरेंद्र मोदी पठानकोट एयरबेस पहुंच गए हैं. वे एयरबेस में सुरक्षा के हालात का जायजा ले रहे हैं और वायुसेनाकर्मियों से मिल रहे हैं. सुबह करीब सवा दस बजे प्रधानमंत्री पंजाब के पठानकोट के लिए रवाना हुए. एयरबेस का जायजा लेने के बाद प्रधानमंत्री बॉर्डर इलाकों का हवाई सर्वेक्षण भी करेंगे. पठानकोट एयरबेस पर पिछले हफ्ते आतंकियों ने हमला किया था. पाकिस्तान से आए आतंकियों के हमले को विफल कर दिया गया था. सभी 6 पाकिस्तानी आतंकी मारे गए थे. 7 सुरक्षाबल भी शहीद हुए थे. भारत ने पाकिस्तान को सबूत सौंपते हुए दोषियों के खिलाफ सख्त कार्रवाई करने को कहा है. जानकारी के मुताबिक, प्रधानमंत्री के साथ आर्मी और एयरफोर्स के चीफ भी मौजूद रह सकते हैं. एयरबेस पर पाकिस्तानी आतंकियों ने हफ्ते भर पहले हमला किया था, जिसमें सात जवान शहीद हुए थे. सुरक्षा बलों ने मुठभेड़ में सभी छह आतंकियों को मार गिराया था, जबकि करीब पांच दिनों तक पूरे इलाके में तलाशी अभियान चलाया गया था.

2) Hindi Text Short Summerisation:

पठानकोट पहुंचे प्रधानमंत्री मोदी एयरबेस का जायजा ले बॉर्डर इलाकों का करेंगे हवाई सर्वे

Various feature extraction methods are available in Hindi text summarization such as rule based, statistical, neural network and hybrid methods. In the proposed system TF-IDF and word2vec are considered as feature extraction methods. TF-IDF comes under statistical feature extraction method and word2vec is the neural network-based feature extraction method. The output of the text feature extraction is sent to CRO based DRNN. Results obtained in Tokenization and feature extraction step is shown in Table II. Sample result obtained in average score generation and final summarized text document is shown in Table III.

TABLE II. SAMPLE RESULT OBTAINED IN TOKENIZATION AND FEATURE EXTRACTION STEP

Document No.	BERT Tokenization	Text Feature Extraction
Doc1	'प्रधानमंत्री', 'के', 'साथ', 'आर्मी', 'और', 'एयरफोर्स', 'के', 'चीफ', 'भी', 'मौजूद', ' रह', 'सकते', 'हैं', 'एयरबेस', 'पर', 'पाकिस्तानी', 'आतंकियों', 'ने', 'हफ्ते', 'भर', 'पहले हमला', 'किया', 'था', 'जिसमें', 'सात', 'जवान', 'शहीद', 'हुए', 'थे', ...	TF:15, IDF: 3252 Cosine similarity between sentence1 and sentences2 using CBOW: 0.999249298413 Skip Gram: 0.885471373104

TABLE III. SAMPLE AVERAGE SCORE AND SUMMARIZED TEXT DOCUMENTS

Document No.	Average Scores	Summarized Text Documents
Doc1	0.76	प्रधानमंत्री मोदी पठानकोट पहुंच गए. एयरबेस में सुरक्षा के हालात का जायजा. बॉर्डर इलाकों का हवाई सर्वेक्षण करेंगे.
Doc2	0.81	सचिन अपने दोहरे शतक भारतीय लोगों के समर्पित करते. तेंदुलकर भारतीय क्रिकेट प्रेमियों का आभार व्यक्त किया.

Document No.	**Average Scores**	**Summarized Text Documents**
Doc3	0.78	आर. पी. एन. सिंह कहा छत्तीसगढ़ में कांग्रेस नेताओं पर हुए नक्सली हमले में सुरक्षा. सुरक्षा खामियां रही हैं मामले जांच एनआईए करेगी.

B. Analysis of performance on evaluation metrics

To draw a confusion matrix for Hindi text document summarization, first need to calculate the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) for the summarization algorithm. In this matrix, the rows represent the actual classification of the sentences (positive or negative), while the columns represent the predicted classification of the sentences (positive or negative). To fill in the cells of the matrix, count the number of sentences that fall into each category based on the true and predicted classifications. If the summarization algorithm correctly identifies 700 important sentences as important and includes them in the summary (TP), while also correctly identifying 228 unimportant sentences as unimportant and excludes them from the summary (TN). Table IV shows the confusion matrix of Hindi text document summarization.

TABLE IV. CONFUSION MATRIX OF HINDI TEXT DOCUMENT SUMMARIZATION

	Predicted Positive	**Predicted Negative**
Actual Positive	700	41
Actual Negeative	31	228

There are 731 important sentences and 269 unimportant sentences in the original document. The summarization algorithm correctly identified 700 of the important sentences (TP) and incorrectly included 41 unimportant sentences in the summary (FP). It also correctly identified 228 of the unimportant sentences (TN) and incorrectly excluded 31 important sentences from the summary (FN). Once the cells of the confusion matrix are filled, use it to calculate various performance metrics such as precision, recall, and f-measure. These metrics can provide insights into the performance of the summarization algorithm and can be used to compare different summarization algorithms. Proposed CRO based DRNN obtained the precision, recall, f-measure of 93.54%,95.1%, 94.31% respectively.

C. Comparative Analysis

Proposed hybrid optimization based Hindi document summarization is compared with existing summarization methods such as MCRMR_SSO, graph based_PSO summarization, genetic algorithm and PEHO based DLSTM. The comparative analysis of CRO based DRNN method with the existing text summarization methods are shown in Table V.

TABLE V. COMPARATIVE ANALYSIS

Metrics	**MCRMR-SSO**	**Graph based-PSO**	**Genetic Algorithm**	**PEHO DLSTM**	**CRO DRNN**
Precision	83.19	89.54	91.56	91.97	93.54
Recall	83.64	89.54	91.56	94.03	95.1
F-Measure	86.06	88.03	90.04	92.97	94.31

The precision attained by the existing techniques, such as MCRMR-SSO is 83.19%, Graph-based_PSO approach is 89.54%, Genetic Algorithm is 91.56%, PEHO based DLSTM is 91.97% whereas the proposed CRO based DRNN achieved the higher performance of precision, which is 93.54%. The obtained recall measure by the proposed CRO base DRNN is 95.1% and the conventional approaches, like MCRMR-SSO, Graph-based_PSO approach, Genetic Algorithm and PEHO based DLSTM attained the recall measure of 83.64%, 89.54%, 91.56% and 94.03%, respectively. The proposed CRO based DRNN achieved the higher performance for f-measure value 94.31%, whereas the existing methods, such as MCRMR-SSO, Graph-based_PSO approach, Genetic Algorithm and PEHO based DLSTM approach attained the f-measure of 86.06%, 88.03%, 90.04% and 92.97%, respectively. Fig. 2 illustrates the performance of existing text summarization techniques with proposed system CRO based DRNN.

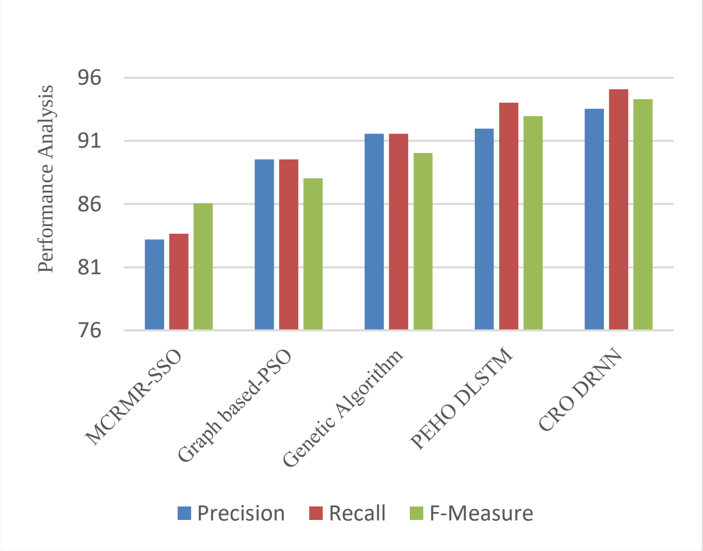

Figure 2. Performance analysis of existing text summarization techniques with CRO based DRNN

V. CONCLUSION

In this research work addresses how the summary of the Hindi documents is generated using hybrid optimization based DRNN and BERT tokenization. The DRNN is employed to create the process for generating sentence scores, and the DRNN is trained using the CRO algorithm to improve the training

Article Received: 25 March 2023 Revised: 30 April 2023 Accepted: 12 June 2023

procedure. The CRO algorithm is produced by combining COOT and ROA. Here, the performance improvement of the CRO-based DRNN is compared to that of other contemporary traditional text summarization methodologies, such as MCRMR-SSO, Graph-based-PSO, Genetic Algorithm, and PEHO-based DLSTM, to determine the method's superiority. Additionally, the precision, recall, and f-measure of the CRO-based DRNN method's performance are analyzed separately for each of the three evaluation criteria. According to the analysis, the CRO-based DRNN method performed better than other conventional methods, achieving higher values for precision, recall, and f-measure of 93.54%, 95.1%, and 94.31%, respectively. By merging the newly proposed technique with existing optimization-based techniques, the overall system can be further examined and improved in the future.

REFERENCES

[1] Pareek, G., Modi, D. and Athaiya, A., “A Meticulous Approach for Extractive based Hindi Text Summarization using Genetic Algorithm”, 2017.

[2] Vimal Kumar, K. and Yadav, D., “An improvised extractive approach to Hindi text summarization”, In proceedings of Information systems design and intelligent applications, pp. 291-300, Springer, New Delhi, 2015.

[3] Bafna, P.B. and Saini, J.R., “Hindi multi-document word cloud-based summarization through unsupervised learning”, In proceedings of 2019 9th International Conference on Emerging Trends in Engineering and Technology-Signal and Information Processing (ICETET-SIP-19), pp. 1-7. IEEE, November 2019.

[4] Gupta, M. and Garg, N.K., “Text summarization of Hindi documents using rule based approach”, In proceedings of 2016 international conference on micro-electronics and telecommunication engineering (ICMETE), pp. 366-370, IEEE, September 2016.

[5] Thaokar, C. and Malik, L., “Test model for summarizing hindi text using extraction method”, In proceedings of 2013 IEEE Conference on Information & Communication Technologies (pp. 1138-1143), IEEE, April 2013.

[6] Yao, J.G., Wan, X. and Xiao, J., “Recent advances in document summarization”, Knowledge and Information Systems, vol.53, no.2, pp.297-336, 2017.M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

[7] Khurana, A. and Bhatnagar, V., “Investigating entropy for extractive document summarization”, Expert Systems with Applications, vol.187, p.115820, 2022.

[8] Katoch, S., Chauhan, S.S. and Kumar, V., “A review on genetic algorithm: past, present, and future”, Multimedia Tools and Applications, vol.80, no.5, pp.8091-8126, 2021.

[9] Kumar, K.V., Yadav, D. and Sharma, A., “Graph based technique for Hindi text summarization”, In proceedings of Information systems design and intelligent applications, pp. 301-310, Springer, New Delhi, 2015.

[10] Jain, A., Arora, A., Morato, J., Yadav, D. and Kumar, K.V., “Automatic text summarization for Hindi using real coded genetic algorithm”, Applied Sciences, vol.12, no.13, p.6584, 2022.

[11] Rahim Khan, Yurong Qian, and Sajid Naeem; "Extractive based Text Summarization Using K-Means and TF-IDF", International Journal of Information Engineering & Electronic Business, vol.11, no.3, 2019.

[12] Wang, D., Tan, D. and Liu, L., “Particle swarm optimization algorithm: an overview”, Soft computing, vol.22, no.2, pp.387-408, 2018.

[13] Gulati, A.N. and Sawarkar, S.D., “A novel technique for multidocument Hindi text summarization”, In proceedings of 2017 international conference on nascent technologies in engineering (ICNTE), pp. 1-6, IEEE, January 2017.

[14] Dalal, V. and Malik, L., “Semantic graph based automatic text summarization for hindi documents using particle swarm optimization”, In International Conference on Information and Communication Technology for Intelligent Systems, pp. 284-289, Springer, Cham, March, 2017.

[15] Pradeepika Verma; and Hari Om; "MCRMR: Maximum coverage and relevancy with minimal redundancy based multi-document summarization", Expert Systems with Applications, vol.120, pp.43-56, 2019.

[16] Bandari, S. and Bulusu, V.V., “Feature extraction based deep long short term memory for Hindi document summarization using political elephant herding optimization” International Journal of Intelligent Robotics and Applications, pp.1-16, 2022.

[17] Rupal Bhargava, Gargi Sharma, and Yashvardhan Sharma; "Deep text summarization using generative adversarial networks in indian languages", Procedia Computer Science, 167, pp.147-153, 2020.

[18] Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., “Bert: Pre-training of deep bidirectional transformers for language understanding”, arXiv preprint arXiv:1810.04805, 2018.

[19] Liu, Q., Wang, J., Zhang, D., Yang, Y. and Wang, N., “Text features extraction based on TF-IDF associating semantic”, In proceedings of IEEE 4th International Conference on Computer and Communications (ICCC), pp. 2338-2343, December 2018.

[20] Lilleberg, J., Zhu, Y. and Zhang, Y., “Support vector machines and word2vec for text classification with semantic features”, In proceeding of 2015 IEEE 14th International Conference on Cognitive Informatics &Cognitive Computing (ICCI* CC), pp. 136 140, IEEE ,July 2015.

[21] Inoue, M., Inoue, S. and Nishida, T., “Deep recurrent neural network for mobile human activity recognition with high throughput”. Artificial Life and Robotics, vol.23, no.2, pp.173-185, 2018.

[22] Jia, H., Peng, X. and Lang, C., “Remora optimization algorithm”, Expert Systems with Applications, vol.185, p.115665, 2021.

[23] Naruei, I. and Keynia, F., “A new optimization method based on COOT bird natural life model”, Expert Systems with Applications, vol.183, pp.115352, 2021.

[24] Bandari, S. and Bulusu, V.V., “BERT Tokenization and Hybrid-Optimized Deep Recurrent Neural Network for Hindi Document Summarization”, International Journal of Fuzzy System Applications (IJFSA), vol.11, no.1, pp.1-28, 2022.

[25] Hindi Text Short Summarization Corpus data set available at “https://www.kaggle.com/disisbig/hindi-text-short-summarization-corpus”, accessed on December 2022.