Hybrid Approach for Multi-Document Text Summarization by N-gram and Deep Learning Models

S.Sudha Lakshmi
Research scholar, Dept. of Computer Science
SPMVV, Tirupati, India.
s_sudhamca@yahoo.com

Dr.M.Usha Rani
Professor, Dept. of Computer Science
SPMVV, Tirupati, India.
musha_rohan@yahoo.com

ABSTRACT—Text Summarization is a technique in which the source documents are shortened, the relevant information is filtered and a concise version is produced. It gives the gist of the whole document to improve the readability by avoiding redundant information. Multi-document summarization (MDS) is an efficient system that generates a summary by the collection of information from various topic-related documents in clusters. MDS by combining extraction and abstraction schemes are limited and still, it is a challenging research problem. In this paper, the current research work aims to achieve the goal of MDS by a novel hybrid framework called HEATS (Hybrid Extractive and Abstractive Text Summarization) is presented which generates an efficient summary by using the N-Gram model for extractive and RNN-LSTM-CNN deep learning architecture for an abstractive summary. This paper also draws attention towards benchmark datasets such that the proposed system was evaluated on it and its performance is found to be pretty good when compared to existing systems in terms of ROUGE scores. The experimental results show that the proposed framework attains the highest values in terms of ROGUE scores as ROUGE1-39.86, ROUGE 2- 19.72, ROUGE 3-38.03, and ROUGE L-39.38 on DUC2003 and ROUGE1-43.08, ROUGE 2-21.02, ROUGE 3-43.87 and ROUGE L-43.28 on DUC2004.

Keywords- ***Automatic text summarization, Deep learning models, Multi-documents, N-gram, ROUGE scores.***

Table I Nomenclature

Abbreviation	Description
HEATS	**H**ybrid **E**xtractive and **A**bstractive **T**ext **S**ummarisation
ATS	Automatic Text Summarization
NLP	Natural Language Processing
LSTM	Long Short-Term Memory
RNN	Recurrent Neural Networks
CNN	Convolutional Neural Networks
ROUGE	Recall Oriented Understudy for Gisting Evaluation
DNN	Deep Neural Networks
VAE	Variational Auto-Encoder
SDS	Single-Document Summarization
MDS	Multi-Document Summarization
FWC	Frequency Weight Component

1. INTRODUCTION

In the Current Big Data Era, it is increasingly challenging to quickly mine useful information from massive amounts of text data from every possible source. From the readers' viewpoint, gathering the relevant information from voluminous documents is time taking and labour intensive. Automatic Text summarization is one of the important tasks of Natural Language Processing(NLP) that automatically converts a text or a group of texts to extract important semantic information within a similar topic into a brief summary. Usually, the generated summaries are compressed in a manner than the original documents. Single document summarization intention is to generate a summary from only one source document, while multi-document summarization objective is to generate a concise and informative summary across a group of topic-related documents.From the technical point of view, MDS is much more complex to tackle than single-document summarization [1].

Summary evaluation is one more challenging problem in this research field.[2] Due to the wide range of applications in many fields, automatic text summarization got the attention of many researchers. There are two main classes in which automatic text summarization can categorize one is abstractive, and the other is an extractive method [3]. An Extractive scheme considers keywords, phrases, and sentences in model summaries that are selected from the source articles [4] [5] [6]. Abstractive text summarization is a method of generating a precise summary that contains paraphrased sentences or novel phrases that may not be present in the source documents.

MDS requires good analysing capabilities such that models can identify and merge the information from the available corpora. Multi-document summarization systems are extensively used in several real-world applications such as medical documents [1, 7], scientific publications [8], product reviews [8], and Wikipedia articles generation [9], summarization on the news [10]. In the Current Scenario, deep learning models have been applied for multi-document summarization tasks[11,13],which flourishes the development of ATS and empowers the models to achieve improved performance.Lack of sufficient training data, DL models also face the computational issues in processing of multi-documents. [12] Existing systems for multi-document text summarization used several techniques such as Term-Frequency (TF-IDF) Based, Graph-Based (Text Rank), Cluster-Based (K-means, Hierarchical), Latent Semantic Analysis (LSA), and Machine learning techniques. [14, 22].

The main challenges in the MDS system are: to generate summaries that have minimum redundancy and maximum coverage of content. Various deep learning approaches have been applied in text summarization such as supervised models [15] such as seq2seq models, recurrent neural networks, convolution neural networks [13], LSTM, Attention Mechanism, and unsupervised techniques like Auto encoders, variation auto-encoders, RBM, Transformers Etc. However, they are used exclusively for either extractive or abstractive, only for single or multi-document. The hybrid scheme combines both the extractive and abstractive approaches. In MDS, automatic summarization is categorized into two approaches: learning the natural sequence of the sentence from the huge corpora and chronological information, sentence extraction using an extractive approach, and frequency weight adjustment, which is considered input for abstractive approach for the best summary generation. However, hybrid approaches for multi-document text summarization using deep learning models on benchmark datasets are very limited. This motivates the current research work.

The proposed system presents an Automatic Text summarization for MDS to overcome existing summarization systems' drawbacks and challenges that aims to fulfil the identified research gap due to the lack of thorough surveys on multi-document benchmark datasets with a blend of deep learning models or neural networks.

The Proposed Framework consists of the following Research contributions:

- An N-gram model generates an extractive summary by merging the source documents from various clusters related to a topic.
- An extractive summary is considered as Labels and given as an input to the deep learning architecture along with the pre-processed merged document to generate an Abstractive summary.
- A Novel Hybrid framework (HEATS) is proposed to generate Summary by extractive and abstractive methods on benchmark datasets for MDS using semi-supervised DL models. The results are evaluated using ROUGE scores and compared to the existing state-of-art models.

In this Hybrid approach, multiple documents related to a particular topic from various clusters of DUC 2003, DUC 2004 datasets are merged into a single document which is given as input, and then pre-processing like tokenization, stop word removal, contraction mapping was done to remove unnecessary things which are not useful in model summary. The proposed HEATS framework consists of hybrid approach for extractive and abstractive text summarization follows two stages: The first stage, the n-gram based extractive summarization approach, is performed. The generated Summary from this approach is used as a label for the other process, which is given as an input to the DL model and original merged text to generate an abstractive Summary during the training phase. During the testing phase, new multi documents were given as input to the model to obtain final predicted summary.

The rest of the paper is organized as follows:
Section2 Literature survey associated with automatic text summarization approaches was studied.
Section3 Presents the proposed Novel framework and its key components.
Section4 Presents the performance evaluation of the proposed system in comparison to the existing state-of-the-art models and generated models. Finally discusses the results attained.
Section5 Draws the conclusions of the paper and paves a path for future research work.

2. Literature Survey

In the MDS tasks, the length of input documents and their input types varies as short, long, mixed. The quantity of the input data is quite large enough whereas the length of every single document is pretty short. The characteristic representation of this type of input data is product reviews [16]. The next case presents, the input data from source documents is relatively small but the length of each document is long. For instance, a summary from a group of news articles [17], numerous Wikipedia web articles with different styles [18]. Hybrid documents with an amalgamation of several long and short documents. For example, reader-aware MDS comprises of news with several readers' comments. Additional example is making a scientific summary from a lengthy scientific paper with several short corresponding citations for multiple documents [16].

Table II: Survey on Automatic Text summarization

Year	Reference	Framework	Remarks
2019	H. Zhang et al[23]	A Word level encoding method with self-attention [24] mechanism was implemented.	Minimum length of text-only applied for dialogue generation.
	S.Song et al[26]	LSTM-CNN for phrase-based semantics utility was implemented	Eliminates OOV and long text problem but complete sense of the sentence was carried away.
	X. Chenet al[30]	Beam search of word-to-word relation was implemented	Selecting key words, relation between the selected words for headlines. OOV breach problem and trying to solve the grammatical portions. No long text summary generation

ISSN: 2278-4632
Vol-11 Issue-02 2021

2020	J. M. Sanchez-Gomez et al[19]	A BEE colony optimization algorithm was implemented	OOV problem was not solved and helps in generating an extractive summarization
	A. Hernandez-Castaneda[20]	Hierarchal clustering with genetic algorithm were combined to generate summary	A unique word-based summarization with context was observed but not for long texts.
	M. Yang et al[21]	LSTM with guidance vector was implemented to generate abstractive summarization	Computational cost and summary generation time were too heavy.
2021	J. Deng et al[24]	Attention based Bi-LSTM based extractive summary generation was implemented	Long text summary ,can be applied to multiple languages but OOV was not solved
	L. Huang et al[28]	RNN As bi-directional gated linear units as fusion of words for context selection	minimizes redundancy and context missing and OOV but could not be used for large documents and generates only a single sentence as headline.
	P. Li et al[29]	Pointer generator method, Bi-directional LSTM with decoder module only	OOV problem was solved. But only used single line text generation for low vocabulary rate

3. METHODOLOGY

Generalized summarisation models possess extractive and abstractive summarization tools [23, 26] but these approaches results were deviated from one another and not relative to the summaries. Later on some hybrid approaches are evolved for the comparative analysis to raise the accuracy in the system in generating summary. In this novel proposed framework, a hybrid summarization methodology was applied with the more relevant summary generation with reduced redundancy in the sentences and words relevant to the main content by creating a dictionary from the existing information. The entire process of summary generation in this paper was split into two sections. One is extractive summary generation using the N-gram Model, and the other is abstractive summarization using the deep learning models and the combination of this was already mentioned in section 1 as HEATS framework. However, before predicting the summary of a text, the chosen documents need to undergo a pre-processing approach.

3.1 PRE-PROCESSING

Text pre-processing schemes helps in the elimination of redundancy, remove unwanted words, and make text split from paragraph to sentence and sentences to words. The proposed framework comprises of pre-processing steps like tokenization, stop words removal, contraction mapping.

3.2 N-GRAM BASED EXTRACTIVE SUMMARIZATION

In MDS extractive summarization, the selection of important sentences is the main step. For this purpose, the N-gram approach was used to generate an extractive summary which is considered as a label for further process. An N-gram can be termed as a series of N-words such as Uni-grams, Bi-grams, etc. Later the performance of the proposed model was evaluated using ROGUE scores which generously accessed N-grams. The N-Gram approach followed in this paper requires a probabilistic reference model of sentence selection, which considers weight and length as parameters. To estimate each word's probability and generate word pairs (previous word, current word), it uses frequency weight component calculation, and then verifies whether the sentence should be part of summary or not, afterwards the extractive summary is generated.

Algorithm 1: Extractive Summarization Model based on N-grams model

Input: Merged Multi documents, set sentence length, set Start and Stop tokens for each sentence

STEP 1: for all sentences(S) do
STEP 2: Get the length of each sentence using a total count of words in the sentence.
STEP 3: Verify the Sentence length and calculate the Frequency Weight Component of words in each sentence
STEP 4: if the Sentence length parameter is matched or greater than the static word count of each sentence
STEP 5: Select the sentence
STEP 6: else
STEP 7: Update sentence count go to Step 2
STEP 8: end if
STEP 9: end for
Output: Extractive Summary.

ISSN: 2278-4632
Vol-11 Issue-02 2021

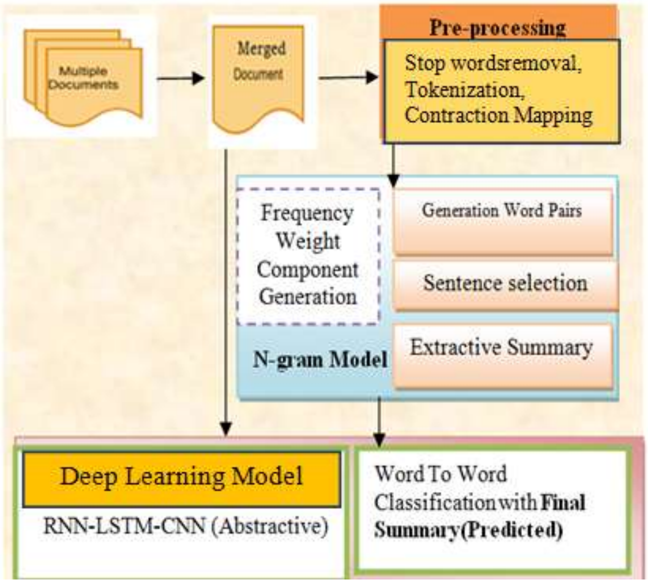

Fig 1: Proposed **HEATS** framework

This paper deals with a little bit of probability. In order to build vocabulary using deep learning models, a huge corpus of trained data is required. Therefore, building an NLP model by predicting the words in a sentence with the probability of a word in a series of words plays a significant role.

Based on the mathematical problem formulation Total merged text (TM_{text}) is given as an input to the system, then tokenization, contraction mapping, and stop word removal were applied as pre-processing approaches on the original merged N-Gram model-based summarization was performed on the merged pre-processed text. Sentence length is considered as a parameter for N-gram approach.

An N-gram is used for sentence selection, and Frequency Weight Component (FWC) was calculated to perform summarization. Therefore, the selection of sentences is given as

$$P_{text}(S) = \prod_{FWC} P(word) \qquad (1)$$

Where $P(word) \in TM_{text}$

Each paragraph eliminates the stop words and outcomes in $P_{text}(S)$ as a pre-processed sentence of the document. The concatenation of P (word) results in new sentences, and for that, frequency weight components for each pre-processed word are calculated. But the data obtained from (1) results in continuous form as it extracted from the sentence, but logically it was discretized one and represented as

$$\log(P_{text}(S)) = \sum_{FWC} \log(P(word)) \qquad (2)$$

The pre-processed stage eliminates the unwanted words from the sentences, results in discretized one and was subjected to likelihood test to get an extractive summary and verifies in P_{text} whether it is present in the Total merged text (TM_{text}) or not using Average log-likelihood (avgll) is given by (3)

$$avgll = \sum_{FWC} \log(P(word)) \cap TM_{text} \qquad (3)$$

The existing approaches used similarity measures to identify the duplication of information from the input documents.

In this proposed hybrid approach, Redundancy elimination was performed due to the limitation of the length in summary sentences based on the likelihood ratio of the text. Then, the extracted words from the original text and verified words obtained after redundancy elimination chain rule were applied to compute the joint probability of words in a sequence.

$$Prob(sentence) = Prob(X1)Prob(X1|X2)Prob(X1|x2|X3)Prob(X1) (4)$$

In equation (4) probability of each word (Prob(X1)) and the probability of word pairs (Prob (X1|X2)) was calculated, and concatenation of word pairs was performed to generate the sentences. Here X1, X2... Xn are words in a sentence.

Then obtain the sentences with exact or more than the length of words required. Next eliminate the sentences that have not reached the maximum length of static word count for new summary generation (GS). Finally, the union of sentences was accessed mathematically for extractive summary generation using the selected sentences.

$$GS = \bigcup_{FWC}^{L=100} \begin{bmatrix} Prob(sentence1); \\ Prob(sentence2) \ldots Prob(sentence) \end{bmatrix} \qquad (5)$$

Then GS (Generated new summary) and TM_{text} (original merged text) are subjected to contraction mapping with frequency weight component calculation based on word count of 100(length of words in summary) as a condition in the selection of sentences. FWC of TM_{text} is given as an input to the deep neural networks and GS as labels (Extractive).In existing approaches, abstractive summary generation does not yield promising results. This approach aims in designing a new DNN architecture comprised of RNN-LSTM-CNN for weight optimization and reducing the dimensionality of the text documents. This kind of architecture is beneficial in the case of varying input sizes and extensively used in text summarization tasks. In this novel hybrid approach, CNN was used for backward propagation which helps in minimizing the error during classification, and LSTM to optimize the memory resources was used in the forward direction to associate the relationship between the original text and labels.

In this approach, a single word is mostly comprised of multiple labels and a single label is mostly comprised of multiple words. Based on the probability of the previous word and the current word, the label will be adjusted. The vocabulary used to generate a dictionary during the RNN-LSTM-CNN training phase, then test document words were verified with the trained vocabulary linearly with individual values and sentences.But the trained data and labels were subjected to weight calculation for word pairs .The frequency can be calculated by the inverse of weight and it can be computed for both original text and labels and their set is given as an input to RNN-LSTM-CNN architecture. For Instance, the words related to Cambodian news are {Cambodian, Leader, Hum, Sen.} for each word weight and frequency (FWC) was calculated.

For the word “Cambodian” Frequency weight component generated as follows:

Initially each letter is connected with ASCII conversion therefore {C, a, m, b, o, d, i, a, n} to {67, 97, 109, 98, 111, 100, 105, 97, 110}. Then calculate mean of the word using letter composition. The resultant weight for the word is 99.33. Then the frequency obtained for the word Cambodian is 0.010067 which is obtained by the inverse of weight. Thus, the weight and frequency for each and every word is calculated for both generated summary words and original text after removal of stop words. After eliminating the stop words, the final words of the sentences were provided as set of words for the text original text. Based on the probability of the previous word and the current word, the label will be adjusted. The vocabulary used to generate a dictionary during the RNN-LSTM-CNN training phase, and then test document words were verified with the trained vocabulary.

3.3 Abstractive text summarization using deep learning architecture

This paper implements a new deep learning architecture with RNN, LSTM and CNN to generate an abstractive summary. Here RNN was used as an input layer to associate the relation between original text sentences and the labels generated from the extractive summary. The RNN mechanism let hidden Markov model predict the work with verifying backwards projected words. This necessity in the implementation made to modify the traditional RNN design. The output obtained from RNN is fed into LSTM, which acts as a varied hidden layer to optimize the memory as the sentence length differs in the processed text. Sentences also can be seen as a sequence of words, and having some memory of previous words has proven to be useful in classification.

CNN acts as an output layer to classify and generate the final summary and verifies the test set sentences are presented or not in the trained set. In the summarization systems, the important phrases from the test data are prominent in the summary, but they may be unseen or infrequent in training data are called out-of-vocabulary (OOV) words. Fortunately, RNN in this architecture is easily conceptualized. As an alternative, implementing an RNN only in the forward mode by selecting the first token to the last token in the sentences, while running from back to front token wise RNN selects words in the reverse direction, which allows the system not to miss the context of the sentence and enhances the chances for better training. RNN is adjusted by a flexible hidden layer for connecting forward and backward operations of LSTM. The sentence and word adjustment with labelling follows the same process in forwarding and backward recursions can be done with dynamic programming of the hidden Markov model. The main objective is to maintain statistical meaning between current and previous words in a sentence. For easy access and interpretations of summary, generic and learnable functions need to be adopted. This transition helps in creating a dictionary from an extractive summary by using the proposed deep learning model. To implement this transition, the generic functions were separated, and also by accessing learnable functions, the layers in the DL architecture get concatenated.

Algorithm 2: Abstractive Summarization Model

INPUT: Text, Label, Start and Stop tokens of each sentence

STEP **1:** for all sentences(S) do
STEP 2: Get words from each sentence of the text document
STEP 3: Set labels with words after pre-processing
STEP 4: Training phase: Label each word in the document and generate a dictionary
STEP 4.1: RNN based word- label relation (Input Layer)
STEP4.2: LSTM based memory optimization (Hidden Layer)
STEP 4.3: CNN based classification – (output layer)
STEP 5: end for
STEP6: Testing phase: selecting sentences from test documents, words form sentences by setting the start and stop tokens.
STEP7: Word and sentence classification from a pre-processed merged document from start and end tokens of a sentence.
STEP 8: Group the labels by replacing words in the test sentences obtained after classification
OUTPUT: Predicted summary.

For step 3, a mini-batch size was given as a kernel input $Xt \in Rn \times d$ (Xt: final label, Rn: Relation between words and labels, n: no. of words, d: a label for each word in the document and ϕ: the hidden layer activation function) was provided. In the proposed architecture, the forward and backward directions for a time step (t) are $H \rightarrow t \in Rn \times h$and $H \leftarrow t \in Rn \times h$correspondingly, where h is the number of hidden units. Thus, the Updating of forward and backward hidden layers are as follows:

H→tH←t=ϕ(XtW(f)xh+H→t1W(f)hh+b(f)h),=ϕ(XtW(b)xh +H←t+1W(b)hh+b(b)h)

Where the weights

$W(f)xh \in Rd \times h, W(f)hh \in Rh \times h, W(b)xh \in Rd \times h,$ And$W(b)hh \in Rh \times h,$

And biases b(f)h∈R1×h and b(b)h∈R1×h for both forward and backward direction are the model parameters.

Further concatenating forward and backward layers using hidden layers H→t and Ht to obtain the hidden state of the word and label relation. Finally, Ht∈Rn×2h is to be fed into the output layer. In deep RNN-LSTM-CNN with multiple hidden layers, information is passed as an input to the next layer. Finally, CNN as an output layer computes the output $Ot \in Rn \times q$ (number of outputs: q): $Ot = HtWhq + bq.$ Here, the weight matrix $Whq \in R2h \times q$ and the bias bq∈R1×q are the model parameters of the output layer. The numbers of hidden layers will be different in both directions.

Algorithm 3: Hybrid Summarization Model

Input: Merged Text, Extractive Summarized text, Start and Stop tokens of each sentence

STEP 1**:** Apply algorithm 1 for Generating Extractive summarization
STEP 2**:**Apply algorithm 2 by replacing the Label using extractive summary
STEP **3:** Extract labels from the dictionary for the words in text or test documents.
STEP **4:** Concatenate the labels obtained after classification to build a final summary.
Output: Final Predicted Summary.

Table III : DNN Architecture

Model Parameters	**Values**
Learning Rate	0.001
Batch size	24
Maximum Iterations	25
Epochs	10-30
Optimizer	Adam

For the implementation of the proposed DNN architecture, Python3.5 with Tensor flow CPU backend, Spider, Anaconda 3platform, NLTK toolkit, NLP libraries tools were used.

ISSN: 2278-4632
Vol-11 Issue-02 2021

While training a model, epochs are termed as a hyper parameter in which the number of times the learning algorithm will work on entire dataset whereas batch size defines the quantity of data used at a given point of time. Adaptive learning rate optimizer algorithm (ADAM) to train deep neural networks.

4. Results and Discussion

The proposed novel HEATS framework works efficiently in terms of performance. Recall-Oriented Understudy for Gisting Evaluation (ROUGE) scoring algorithm is used to evaluate summary quality. Manual evaluation to analyse the quality of the summary is time-consuming and needs a lot of effort. When the system summary matches with more words appearing in the human summary or reference summary results in high ROUGE scores.

4.1 N-gram Co-Occurrence Statistics (ROUGE-N)

Given an N-gram of length n, the ROUGE-N is an evaluation metric that measures the quality of summary between system summary documents against reference document is given by

$$\mathrm{ROUGE}-\mathrm{N_{single}}(\text{system summary}, \text{reference}) = \frac{\sum_{ri\in reference}\sum_{n-gram ri}\text{count}(n-\text{gram}, \text{system summary})}{\sum_{ri\in reference}\text{numNgrams}(r_i)} \quad (6)$$

The elements r_i are sentences in the reference document, Count(n-gram, system summary) is the number of times the specified n-gram occurs in the system summary document, and N-grams(r_i) is the number of n-grams in the specified reference sentence r_i and N stands for the length of the N-gram.

4.2 Longest Common Subsequence (ROUGE-L)

Given a sentence in a document d= $[w_1,\ldots, w_m]$ and a sentence s, where the elements s_i corresponds to words, the subsequence $[w_{i1},\ldots, w_{ik}]$ is a common subsequence of d and s if $w_{i_j}\in\{s_1,\ldots, s_n\}$ for j=1,…,k and $i_1<\cdots<i_k$, where the elements of s are the words of the sentence and k is the length of the subsequence. Thus, the subsequence $[w_{i1},\ldots,w_{ik}]$ is (LCS) the longest common subsequence if the subsequence length k is maximal.

Given a system summary document from set of reference summaries, single one is taken, the union of the longest common subsequence's is given by

$$\mathrm{LCS}\,(,\text{system summary}, \text{reference}) = \bigcup_{ri\in reference}\{W|W\in \mathrm{LCS}(\text{system summary}, r_i)\} \quad (7)$$

LCS (system summary, r_i) is the longest common subsequence's in the system summary document and the sentence r_i from a reference document. Here System Summary is denoted as SS.

The ROUGE-L metric is an F-score measure. To calculate it, first, calculate the recall and precision scores given by

$$R_{score} = \frac{\sum_{ri\in refernce}|SS, ri|}{numWords(reference)} \quad (8)$$

$$R_L(SS, reference) = \frac{(1+\beta^2)R_{score}(SS, reference)P_{score}(SS.reference)}{R_{score}(SS, reference)+\beta^2 P_{score}(SS, reference)} \quad (9)$$

Then, ROUGE-L metric calculates between a system summary document and a reference summary. The benefit of ROUGE-L is that it only requires matching the order of the appearance of words and does not require continuous matching of words. However, it calculates the longest sub-sequence only, both longer and shorter, and ignores the final result value which affects the other candidate sub-sequences. When the respective model achieves higher ROUGE scores then the model shows better performance.

The entire work was carried out on DUC2003 and DUC2004datasets and evaluated on R1 (Unigrams), R2 (bigrams), R3 (trigrams), and RL scores. The DUC 2003 corpus consists of 624 documents and summary pairs whereas DUC 2004 comprises 500 document and summary pairs. Datasets DUC 2003, DUC2004 https:// www. nlpir. nist. gov/projects/duc/data.html DUC dataset published by (NIST) the National Institute of Standards and Technology is an agency of the U.S. Commerce Department. The description of the Dataset is given in Table IV.

Table IV: Dataset Description and Statistics

Dataset Attributes	DUC 2003	DUC 2004
No. of Clusters	30	50
No. of documents in each cluster	8	8
The average number of sentences per document	24	27
Word count in a document	250	250
Maximum Summary length (in words)	100	100

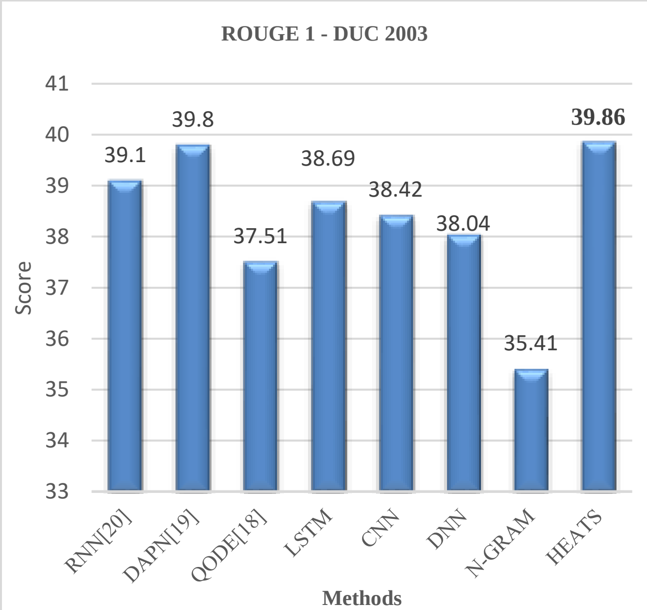

Fig 2: Performance Comparison of HEATS against other models on DUC 2003 based on ROUGE 1

Fig 2 shows the results from eight distinct text summarization (ATS) models and the ROUGE 1 scores were evaluated on DUC 2003 Dataset and their performance was compared with existing and generated models. In this paper, the proposed

ISSN: 2278-4632
Vol-11 Issue-02 2021

HEATS framework provides more relevant scores of the original text in ROGUE-1. Here, the models shown in Fig 2 are traditional deep learning models supervised and unsupervised models while the proposed HEATS framework is a combination of N-gram and RNN-LSTM-CNN models which yields highest scores obtained against the existing deep learning models mentioned .Comparing with eight existing models the proposed HEATS approach was varied from 0.06 to 4.45 score level. The model generates summary with more accuracy after hybrid approach with N-grams model. In ROUGE 1 score the proposed HEATS achieves a highest of 39.86 which is higher than DAPN [19] and RNN [20]

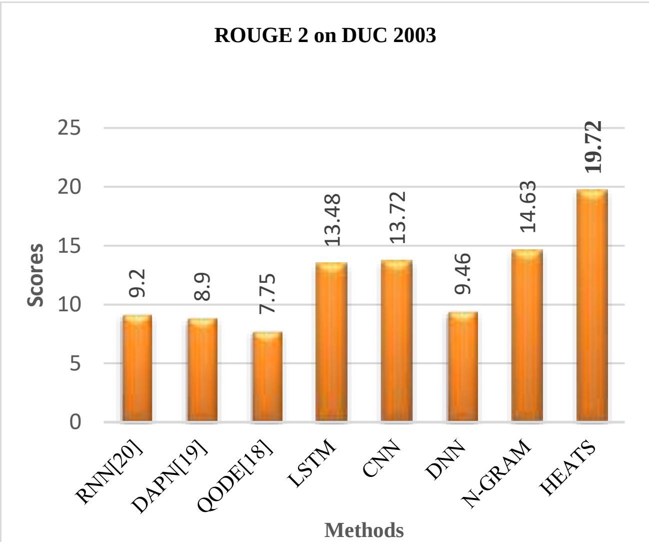

Fig3: Performance Comparison of HEATS between other models on DUC 2003 based on ROUGE 2 scores

Fig3 shows the results from eight distinct text summarization (ATS) models and the ROUGE 2 scores were evaluated on DUC 2003 Dataset and their performance was compared with existing and generated models. The proposed HEATS framework provides more relevant scores of the original text in ROGUE-2.Comparing with existing models against the proposed HEATS approach results varies from 5.09 to 11.07 score level on eight different methods. HEATS framework achieve highest ROUGE 2 of 19.72.

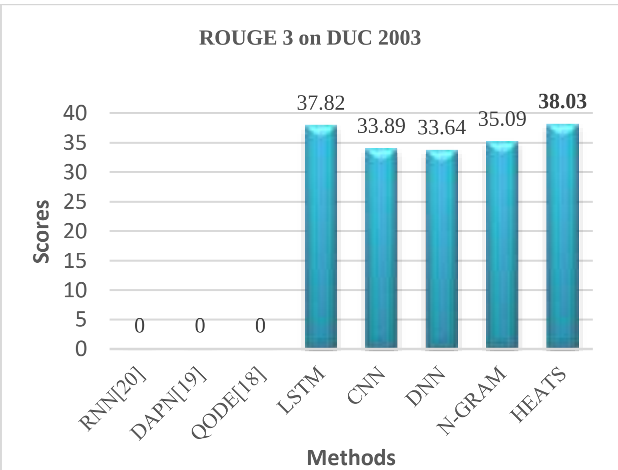

Fig 4: Performance comparison of HEATS between other models on DUC 2003 based on ROUGE 3 scores

Fig4 shows the results from eight distinct text summarization (ATS) models and the ROUGE 3 scores were evaluated on DUC 2003 Dataset and their performance was compared with existing and generated models.The proposed HEATS framework provides more relevant scores of the original text in ROGUE-3.Comparing existing models to the proposed HEATS approach, results varies from 0.21 to 4.39 score level against eight different methods.

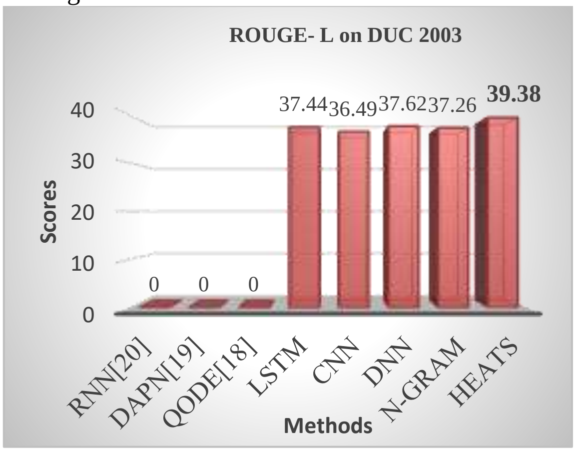

Fig 5: Performance Comparison between proposed system other models on DUC 2003 using ROUGE-L scores

Fig 5 shows the results from eight distinct text summarization (ATS) models and the ROUGE L scores were evaluated on DUC 2003 Dataset and their performance was compared with existing and generated models. The proposed HEATS framework provides more relevant scores of the original text in ROGUE-L.Comparing existing models to the proposed HEATS approach results vary from 1.76 to 2.89 score level against eight different methods.

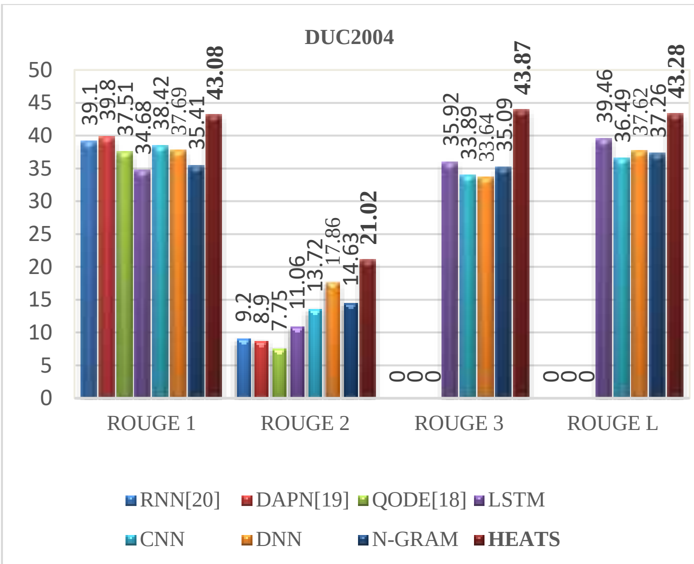

Fig6:Performance Comparison of proposed HEATS framework against other models on DUC2004 based on ROUGE Scores.

Fig 6 shows the results from eight distinct text summarization (ATS) models and the ROUGE scores were evaluated on DUC 2004 Dataset and their performance is compared with existing and generated models. The summary quality and the accuracy on DUC2004 dataset shows that the proposed model shows best results in terms of ROUGE scores relevant to the original text on Various Existing and generated models i.e. ROUGE-1 score 43.08, ROUGE-2 score 21.02, ROUGE-3 score 43.87, and ROUGE-L score are 43.28 respectively.
The above experimental results on benchmark Datasets DUC2003 and DUC2004 for Multi-documents on ROUGE-1, ROUGE-2, ROUGE-3, ROUGE-L justifies that the proposed HEATS model produces better results than the state-of-the-art models of summarization (ATS) concerning semantic and syntactic structure.
To further evaluate the eminence of the summaries, Quality evaluation linguistic aspects of the summary are considered such as Redundancy, Fluency and informativeness. The redundancy indicator measures whether the summary contains repeated information or not. The informativeness indicator can reflect whether the summary covers relevant points from the original merged documents. The fluency indicator emphases on whether the summary is well-formed and grammatical. The qualitative Evaluation of the generated summary on DUC 2003, DUC 2004 datasets for different methods are presented in Table V and Table VI based on factors Redundancy, Informativeness and fluency were calculated mathematically and given as

$$\mathrm{Redundancy} = \mathrm{Count}\left(\frac{\sum(\mathrm{W_i(SS)} \cap \mathrm{W_{i-1}(SS)})}{\text{Total words in SS}}\right) \quad (10)$$

Where W_i are words from System Summary.

$$Informativeness = \frac{\sum(\mathrm{W_{oi}(SS)} \cap \mathrm{W_i(SS)})}{\text{Total words in SS}} \quad (11)$$

Where W_{oi} original text synonym or original word.

$$\mathrm{Fluency} = \frac{\sum(\mathrm{W_{oi}(SS)} \cap \mathrm{W_i(SS)})}{\text{Total words in Original text}} \quad (12)$$

Table V: DUC 2003 (For 100 words of abstract)

Method	**Redundancy**	**Informativeness**	**Fluency**
LSTM	19	68	49
CNN	27	53	46
DNN	33	59	76
N-GRAM	34	69	89
HEATS (Proposed)	12	78	83

In Table V, the redundancy indicator presents that proposed system shows only 12 redundant words in 100 words in DUC 2003,hence the HEATS avoids repetition in sentences. Informativeness indicator specifies, HEATS gives relevant information from original merged text. Fluency factor indicates 83 words out of 100 are fluent means summary is readable, but N-grams model achieves highest in fluency.

Table VI: DUC 2004 (For 100 words of abstract)

Method	**Redundancy**	**Informativeness**	**Fluency**
LSTM	16	68	46
CNN	27	53	42
DNN	24	59	62
N-GRAM	34	69	89
HEATS	14	73	83

In Table VI, the redundancy factor indicates that proposed system shows only 14 redundant words appear in 100 words in DUC 2004 Dataset, hence the HEATS avoids repetition in sentences. Informativeness indicator specifies 73 informative words from 100 words, HEATS gives relevant information from original merged text. Fluency factor indicates 83 words out of 100 are fluent means summary is readable, but N-grams model achieves highest in fluency.

In Table V and Table VI, an abstract of 100 words were observed and measured for redundancy check, informativeness and fluency by Human evaluation. As the stop words were removed, the transformed data was difficult to read. Therefore, a sentence corrector was adopted later and these parameters were observed. The proposed scheme produces better results than the existing ones with a perfect read of 83 words. The average results were tabulated based on the original text and it varies between 68 to 94.

5. Conclusion And Future work

The Proposed Hybrid Extractive and Abstractive summarization(HEATS) Framework for multi-documents using N-gram and deep learning models i.e., RNN-LSTM-CNN accomplishes better performance to other existing state-of-the-art models with regard to syntactic and semantic coherence. Experimental results show that HEATS Framework substantially outperforms various multi-document summarization (MDS) baselines and achieves state-of-the-art models on Benchmark datasets. Results presented based on ROUGE-1(R1), ROUGE-2 (R2), ROUGE-3 (R3) and ROUGE-L (RL) scores on dataset DUC 2003 are 39.86, 19.72, 38.03, and 39.38. The results presented based on ROUGE-1, ROUGE-2, ROUGE-3 and ROUGE-L scores on DUC 2004 as 43.08, 21.02, 43.87 and 43.28.Qualitive evaluation metrics such as Redundancy, Informativeness and Fluency was also performed. Numerically proposed model results in 25% efficiency than the existing approaches. Though proposed framework overcomes few problems of existing models such as summary generation of lengthy texts and OOV problem, it also has few drawbacks. The proposed model works less proficiently in generating labels for effective summary .In future work the model will incorporate BERT or fine tune BERT or any other pre-trained language models along with categorization for Hybrid MDS to improve the overall performance of the Summary.

References

[1] Ma, Congbo, W. Zhang, Mingyu Guo, Hu Wang and Quan Z. Sheng. “Multi-document Summarization via Deep Learning Techniques: A Survey.” 2020.
[2] Gambhir, Mahak and Vishal Gupta. “Recent automatic text summarization techniques: a

survey." Artificial Intelligence Review, Vol 47, 2016.

[3] Sohail, Asim & Aslam, Uzair & Tariq, Hafiz & Jayabalan, Manoj. "Methodologies and Techniques for Text Summarization: A Survey". Journal of Critical Reviews ISSN- 2394-5125 Vol 7, Issue 11, 2020.

[4] Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, SaeidSafaei, Elizabeth D. Trippe, Juan B. Gutierrez and KrysKochut, "Text Summarization Techniques: A Brief Survey", International Journal of Advanced Computer Science and Applications(IJACSA), 8(10), 2017.

[5] N. Bhatia and A. Jaiswal, "Automatic text summarization and it's methods - a review," 2016 6th International Conference - Cloud System and Big Data Engineering (Confluence), 2016, pp. 65-72.

[6] Verma, Rakesh & Lee, Daniel. "Extractive Summarization: Limits, Compression, Generalized Model and Heuristics. Computación y Sistemas". 2021.

[7] Afzal, Muhammad et al. "Clinical Context–Aware Biomedical Text Summarization Using Deep Neural Network: Model Development and Validation." Journal of Medical Internet Research 22, 2020.

[8] M. Yasunaga, "ScisummNet: A Large Annotated Corpus and Content-Impact Models for Scientific Paper Summarization with Citation Networks", AAAI, vol. 33, no. 01, pp. 7386-7393, Jul. 2019.

[9] Liu, Peter J. et al. "Generating Wikipedia by Summarizing Long Sequences." ArXiv abs/ 1801. 10198, 2018.

[10] Fabbri, Alexander R. et al. "Multi-News: A Large-Scale Multi-Document Summarization Dataset and Abstractive Hierarchical Model." ACL, 2019.

[11] Li, Piji et al. "Cascaded Attention based Unsupervised Information Distillation for Compressive Summarization." EMNLP, 2017.

[12] Liu, Yang and Mirella Lapata. "Hierarchical Transformers for Multi-Document Summarization." ACL, 2019.

[13] B. Jang, M. Kim, G. Harerimana, S.-U. Kang, and J. W. Kim, ''Bi-LSTM model to increase accuracy in text classification: Combining Word2vec CNN and attention mechanism,'' Appl. Sci., vol. 10, no. 17, p. 5841, Aug. 2020.

[14] Y. Guan, S. Guo, R. Li, X. Li, and H. Zhang, ''Frame semantics guided network for abstractive sentence summarization,'' Knowl.-Based Syst., vol. 221, Jun. 2021, Art. no. 106973.

[15] Atif Khan, Muhammad Adnan Gul, Mahdi Zareei, R. R. Biswal, Asim Zeb, Muhammad Naeem, Yousaf Saeed, Naomie Salim, "Movie Review Summarisation Using Supervised Learning and Graph-Based Ranking Algorithm", Computational Intelligence and Neuroscience, vol. 2020, Article ID 7526580, 14 pages, 2020

[16] Dima Suleiman, Arafat Awajan, "Deep Learning Based Abstractive Text Summarisation: Approaches, Datasets, Evaluation Measures, and Challenges", Mathematical Problems in Engineering, vol. 2020

[17] Sarker Abeed, Yang Yuan-Chi, Al-Garage Mohammed Ali, Abbas Aamir, "A Light-Weight Text Summarisation System for Fast Access to Medical Evidence", Frontiers in Digital Health, vol.2, YEAR 2020.

[18] Jiang Z, Liu M, Yin Y, Yu H, Cheng Z and Gu Q. Learning from Graph Propagation via Ordinal Distillation for One-Shot Automated Essay Scoring Proceedings of the Web Conference 2021, (2347-2356)

[19] J. Jiang et al., "Enhancements of Attention-Based Bidirectional LSTM for Hybrid Automatic Text Summarization," in IEEE Access, vol. 9, pp. 123660-123671, 2021.

[20] Ramesh Nallapati, FeifeiZhai, and Bowen Zhou. 2017. "SummaRuNNer: a recurrent neural network based sequence model for extractive summarization of documents". In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, AAAI Press, 3075–3081.

[21] J. M. Sanchez-Gomez, M. A. Vega-Rodríguez, and C. J. Pérez, ''A decomposition-based multi-objective optimization approach for extractive multi-document text summarization,'' Appl. Soft Comput., vol. 91, Jun. 2020.

[22] A. Hernandez-Castaneda, R. A. Garcia-Hernandez, Y. Ledeneva, and C. E. Millan-Hernandez, ''Extractive automatic text summarization based on lexical-semantic keywords,'' IEEE Access, vol. 8, pp. 49896–49907, 2020.

[23] M. Yang, X. Wang, Y. Lu, J. Lv, Y. Shen, and C. Li, ''Plausibility-promoting generative adversarial network for abstractive text summarization with multi-task constraint,'' Inf. Sci., vol. 521, pp. 46–61, Jun. 2020.

[24] H. Zhang, Y. Lan, L. Pang, J. Guo, and X. Cheng, ''Detecting the relevant contexts with self-attention for multi-turn dialogue generation,'' in Proc. 57th Annu. Meeting Assoc. Comput. Linguistics, Florence, Italy, 2019, pp. 3721–3730.

[25] J. Deng, L. Cheng, and Z. Wang, ''Attention-based BiLSTM fused CNN with gating mechanism model for Chinese long text classification,'' Comput. Speech Lang., vol. 68, Jul. 2021.

[26] S. Song, H. Huang, and T. Ruan, ''Abstractive text summarization using LSTM-CNN based deep learning,'' Multimedia Tools Appl., vol. 78, no. 1, pp. 857–875, Jan. 2019.

[27] M. Sangiorgio and F. Dercole, ''Robustness of LSTM neural networks for multi-step forecasting of chaotic time series,'' Chaos, Solitons Fractals, vol. 139, Oct. 2020.

[28] L. Huang, W. Chen, Y. Liu, S. Hou, and H. Qu, ''Summarization with self-aware context selecting mechanism,'' IEEE Trans. Cybern., early access, Jan. 11, 2021, doi: 10.1109/TCYB.2020.3042230.

[29] P. Li, J. Yu, J. Chen and B. Guo, "HG-News: News Headline Generation Based on a Generative Pre-Training Model," in IEEE Access, vol. 9, pp. 110039-110046, 2021.

[30] X. Chen, J. Li and H. Wang, "Key phrase Enhanced Diverse Beam Search: A Content-Introducing Approach to Neural Text Generation," in IEEE Access, vol. 7, pp. 72716-72725, 2019.