How to Cite:
Lakshmi, S. S., & Rani, M. U. (2022). Hybrid multi-document text summarization via categorization based on BERT deep learning models. *International Journal of Health Sciences*, *6*(S1), 5346–5369. https://doi.org/10.53730/ijhs.v6nS1.6095

Hybrid multi-document text summarization via categorization based on BERT deep learning models

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

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

***Abstract*---**Text summarization is the process of employing a system to shorten a document or a collection of documents into brief paragraphs or sentences using various approaches. This paper presents text categorization using BERT to improve summarization task which is a state-of-the-art deep learning language processing model that performs significantly better than all other previous language models. Multi-document summarization (MDS) has got its bottleneck due to lack of training data and varied categories of documents. Aiming in this direction, the proposed novel hybrid summarization B-HEATS (Bert based Hybrid Extractive Abstractive Text Summarization)framework is a combination of extractive summary via categorization and abstractive summary using deep learning architecture RNN-LSTM-CNN to fine-tune BERT which results in the qualitative summary for multiple documents and overcomes out of vocabulary (OOV). The output layer of BERT is replaced using RNN-LSTM-CNN architecture to fine tune which improves the summarization model. The proposed automatic text summarization is compared over the existing models in terms of performance measures like ROUGE metrics achieves high scores as R1 score 43.61, R2 score 22.64, R3 score 44.95 and RL score is 44.27 on Benchmark DUC datasets. The results shows that B-HEATS framework catches up with variations of summary styles with respect to different categories.

***Keywords*---**Text Summarization, Category_id Score based categorization, BERT, Deep Learning.

Introduction

Automatic text summarization is a subfield of Natural Language Processing (NLP) and text mining that aims to produce a condensed version of one or multiple input documents by extracting the most relevant content. [1] An important branch of NLP [17] [22] [23] [24] [25] is the Automatic text summarization that intends to describe the extended text documents compactly, such that the end-users can quickly read and understand the information. Automatic text summarization plays a pivotal role in a wide range of real-world applications such as the summarization of news, clinical summaries, and scientific publications. In this era, the internet is being a knowledge source for humans, and the textual data generated by it is massive [3] [9] [10] [11]. This approach tends to be more crucial and challenging since the consumers will not be interested in reading long texts. This approach necessitates a novel tool to characterize the content in a concise form called summary. Different from single-document summarization, multi-document summarization (MDS) aims to effectively integrate key information from multiple text documents into a concise and comprehensive report [2]. MDS supervised learning systems, requires large amounts of labeled training data and Manual summaries are extremely labor-intensive and time-consuming process. For Example, generic multi-document summarization benchmark DUC datasets contain less than 400 human reference summaries in whole. The text summarization techniques are grouped as extractive and abstractive text summarization [3] [12] [13][14] [15] [16]. In general extractive approaches contain salient pieces of text, e.g. words, phrases or sentences are identified and taken as the summary, and abstractive approaches, most of which rely on neural methods.[2] Text categorization is a significant research problem in the domain of NLP, which is widely applied in several areas such as pattern recognition, data mining, emails, search engine etc. Most of the research works uses text classification methods which have been previously proposed are based on machine learning, decision tree, k-nearest neighbor, support vector machine, CNN etc. This paper presents BERT state of art deep learning model which is pre-trained for large scale text data, combining word representations and sentence representations in a large transformer which can be fine tune for task specific applications.

Table I
Summary approaches with remarks

Dataset and Approach	Remarks
Dataset: Amazon Fine Food review Approach: LSTM, Deep Learning with attention mechanism Summarization Type: Abstractive	This research was focused on generating abstractive summary on large volume of data using sequential approach. But this approach cannot be applied on large volume of text and even for training lots time needs to hold. [27]
Dataset: DUC 2004 Approach: Neural network language model, local attention mechanism Summarization Type: Abstractive	This approach clears the long text summarization for MDS. But was lag in solving OOV problem and the performance was ruled out compared with existing approaches. [28]

Dataset: DUC 2005-2007 Approach: Bi-LSTM Summarization Type: Abstractive	This paper compares 6 different deep learning approaches and finally Bi-LSTM with max pooling results in best outcomes. But here a query based was generated and no limitation of summary words. [21] [29]
Dataset: DUC 2002, DUC 2003, DUC 2004, DUC 2005, DUC 2006, DUC 2007 Approach: Sentence Relation, Deep Learning with attention mechanism Summarization Type: Extractive	This model presented in the worked on MD datasets to prove their model results in better performance. But was not useful to solve OOV problem.[30]
Dataset: Twitter Data Approach: Word embedding Summarization Type: Extractive	Based on the word concept analysing twitter data and generating summary. But lexicon deep learning approach could not helpful for long text summary generation[31]
Dataset: WebAP Approach: Learning-to-rank, Community Question Answering. Summarization Type: Extractive	A query based summary was generated on basis of answer generation was implemented in this paper. But this approach couldn't control the summary word count and for a large question the answers will be irrelevant.[32]

Summary style differs in several categories. For instance, in DUC datasets, i.e., Politics and Health, state of emergency means information related to countries information, reasons for declaring in Politics. In Health category the summary needs medical emergency such as the patient details, Doctors details also. So summaries should emphasis on diverse aspects of the topics which belong to the categories respectively. To build effective summarization systems, the model should train on better text representations learned by categorization data Moreover, if the category of a document is known in advance, it helps in generating informative summary. BERT (Bidirectional Encoder Representations from Transformers) is one of the best options for a task like summarization and it is a game changer in available NLP models. The major contribution of this research work is:

- A category_id score is introduced for Category identification.
- An extractive summary is generated via categorization identification using the BERT model by mapping the categories.
- Extractive summary via categorization is considered as labels and given as input along with original text for RNN-LSTM-CNN architecture to generate an abstract summary.
- The summary of BERT is fine-tuned via RNN-LSTM-CNN architecture to generate the final summary.

The hybrid framework B-HEATS i.e. BERT based hybrid extractive via category identification and abstractive summary generates final summary reluctantly delivered the best summary compared to the existing Deep learning models.
The rest of the paper is organized as: Section II provides a review of the most recent works in literature. Section III portrays the proposed Automatic text

summarization with an category identification approach. The pre-processing and category identification via category_id score are depicted in Section IV. Further, text mapping to contextualized embeddings via BERT, sentence clustering and sentence selection are addressed in Section V. The results acquired are discussed in a comprehensive manner in Section VI.

Literature Review

Related works

In 2019, Moradi et al. [1] have generated a novel summarization method based on the contextualized embeddings by the BERT model. The BERT-based biomedical summarizer encapsulates four major phases, “pre-processing, mapping text to contextualized embeddings, sentence clustering, and sentence selection”. The input documents identified the informative sentences and the most relevant sentences by combining the clustering method and the different versions of BERT. The proposed summarizer was evaluated against several methods in the literature using the ROUGE toolkit.

In 2020, Zhao et al. [2] had developed SummPip for “multi-document summarization”. The original documents were transformed into a sentence graph in the proposed method by considering both the deep and linguistic representation. The authors have acquired the multiple clusters of sentences by deploying the spectral clustering, and the final summary was generated by compressing each cluster. The resultant of the proposed model had exhibited its enhancement over existing models in terms of consistency and reliability with “DUC-2004 datasets”.

In 2019, Joshi et al. [3] have projected SummCoder for GETS of single documents, and here the authors have generated the summary concerning three selected sentences metrics, namely relevance of the sentence position, novelty of sentence and relevance of the sentence content. In addition, with the deep auto-encoder network and in a distributed semantic space, the similarity among sentences in the form of embeddings was exploited by deriving the novelty metrics. The three-sentence selection metrics were fused and based on their final score. The document summary was generated. A new summarization benchmark, the TIDSumm dataset, was generated to evaluate the proposed work.

In 2019, Anand and Wagh [4] proposed ASTR for supervised DNN, reinforcement learning, and unsupervised PGM based deep learning for representing the aspect/sentiment-aware review. The proposed approach was a multi-task learning system that had merged two major objectives: “domain classification (auxiliary task) and abstractive review summarization (primary task)”. In addition, the authors have proposed a weakly supervised LDA model intending to gather knowledge on sentiment lexicon and domain-specific aspect representations, which were fed as input to the hidden neural states for aspect/sentiment-aware review representations.

In 2018, Qasem A. Al-Radaideh and Dareen Q. Bataineh [5] have developed a hybrid single document approach (ASDKGA) which incorporates domain

knowledge, statistical features and genetic algorithm to extract important points. Further, the approach is tested on KALIMAT and EASC. Finally, the proposed approach is compared with ROUGE.

In 2019, Song et al. [6] introduced ATS for summary sentence creation from different source sentences and preserved the shorter representation with no loss in information. Furthermore, the projected LSTM-CNN based ATS framework (ATSDL) could construct new sentences by exploiting the semantic phrases that were more fine-grained fragments than sentences. Finally, the proposed work has shown reliable results from CNN.

In 2018, Alamiet al. [7] had constructed a new approach for Traditional ATS using the VAE model to learn the feature space from the "high-dimensional input data". Based on the latent representation generated from VAE, the authors have ranked the sentences. The query-based and graph-based approaches were utilized for investigating the impact of the presented work. In 2020, Tomer and Kumar [8] had introduced a novel hybrid approach for GATS. The proposed work was the amalgamation of FLS for extractive sentence selection with Bi-LSTM for abstractive summary production. Moreover, the network weights were updated using the attention mechanism and Adam optimizer. They have explored the most relevant sentences from the document, fuzzy measures and inference. Finally, an abstractive summary was produced for the significant sentences by feeding the relevant sentences as input to Bi-LSTM.

Review

Table 2 shows the features and challenges of existing text summarization approaches. The BERT [1] improves the performance of biomedical text summarization, and here there is a need for decreasing the over fitting or lack of generalization. The SummPip in [2] produces consistent and complete summaries. The fluency, consistency, convergence and redundancy can be improved. SummCoder [3], the processing speed is higher. However, Computational cost is higher.

Further, ASTR in [4] can generate better sentiment-aware summarization. As a result, the computational cost can be reduced. The ASDKGA approach in [5] can reduce perusing time. On the other hand, the loss of synonym detection needs to be considered. LSTM-CNN in [6] improves the effect of phrase acquisition. Apart from this, the proposed approach holds good for smaller datasets. The VAE in [7] learns data from a richer latent space with reduced dimensions. For better accuracy, the noise in the input data needs to be lessened.

Further, Bi-LSTM in [8] can pick out specific elements selectively from that sequence. However, the time of training is higher. Therefore, the need for automatic text summarization is an urgent requirement.

Table II
Features and Challenges of existing works

Author[Citation]	Adapted Methodology	Features	Challenges
Moradi *et al.* [1]	BERT	• Low computational complexity • Shown high correlations with informativeness scores	• Need to investigate the impact of model size • Need to decrease the over fitting or lack of generalization
Zhao *et al.* [2]	SummPip	• Produces consistent and complete summaries • Produces high-quality summaries	• The fluency, consistency, convergence and redundancy can be improved. • Need ranging approach for better reliability
Joshi *et al.* [3]	SummCoder	• Lower errors • Higher processing speed	• Higher Computational cost
Anand and Wagh [4]	ASTR	• Can generate better sentiment-aware summarization	• The computational cost can be reduced.
Dareen and Quasem[5]	ASDKGA	• Reduces perusing time • Determination procedure simple	• Lack of synonym detection • Does not able to summarize multiple document
Song *et al.* [6]	LSTM-CNN	• The phrase acquisition effect is reduced. • The phrase redundancy is reduced	• Time consuming • Requires higher training applicable for smaller datasets
Alami*et al.* [7]	VAE	• Lower training time • Reduces the dimensionality of the matrix	• Content overlap need to be reduced • The noise in the input data need to be lessened
Tomer and Kumar [8]	Bi-LSTM	• Exhibits flexibility • Easy to implement and integrate • Summary is generated with larger dataset	• Time of training is higher

Methodology

In this research work, the proposed novel Hybrid framework B-HEATS is used to generate summary for MDS. The proposed framework consists of the following stages: "Pre-processing, categorization, mapping text via BERT and generated extractive summary as labels, Fine-tune BERT using deep learning for abstractive summary then it generates final summary. The overall architecture of the proposed summarization approach is illustrated in Fig.1. Initially, the collected

documents are pre-processed, and the abstract "categories" that occur are found using a category_id score approach ($D_{category}$).

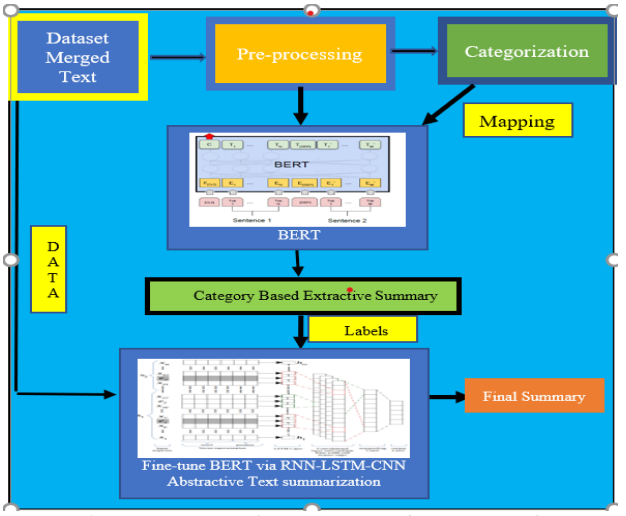

Fig 1.Proposed B-HEATS frame work

Pre-processing and Category Identification using Category_id Score

The text summarization with category identification approach is introduced in this work with following steps.

Pre-Processing

This is the fundamental step and here the unnecessary parts are discarded from D_{input} which is original documents. In general, a document contains signs, symbols, and non-English letters that might not possess information; therefore, eliminating stop words and stemming the words results in pre-processed data. In the case of DUC dataset, the collected data are converted to lower case to remove the ambiguity of words and normalization is done. These parts are considered unnecessary as they do not appear in the “model summaries” for evaluating summary quality. Based on the input text structure and the “user’s preferences”, the customization of the elimination stage takes place. The pre-processing stage further proceeds through splitting into an individual sentence, from which the extraction of tokens determined the categories. Then, with the Natural Language Tool Kit (NLTK), the main text is split into a set of sentences to generate summary. The words (tokens) in each of the sentences will be grouped as the set of tokens T, which are extracted by a novel category_id score calculation.

Category identification using Category_id Score

The Category identification approach comprise of words, sentences and documents. In a document, the "Word" is referred as a fundamental unit of discrete data and document (d) is the organization of N words. The Category is nothing but the distribution of fixed vocabulary. In the dictionary, each of the documents has its own proportions of pre-labelled categories which contain

original text and labels associated with each other. Managing the documents into a related content and divides them based on the category related to it is known as categorization of documents. The traditional Category identification along with summary generation approaches is time consuming and thus making the model more complex. They may fail to analyse the data more accurately, while large datasets are taken into consideration and misleads the context of the sentence. Therefore, in this research work novel category_id score-based category identification with context aware summary was introduced, which makes the text summarization more precise and more informative. In a set of sentences S, the category identification is made based on category_id score i.e. Id1, Id2, Id3 for each sentences$_j$using all tokens T1, T2, T3, …Tn. For instance, the frequency of the token T1 –"Cambodian" can be calculated as no of times it appears in a sentence and it is encoded by all three category_id for a specified category. Similarly for other tokens such as government and opposition etc. The procedure for token specified category identification is shown below:

(a) The token T (token) for all documents distribution α is considered.
(b) Set of sentences S in the document d is assigned to one of the tokens T
(c) For each sentence s_j in d

Compute category_id score Id_1, Id_2, Id_3 as shown below:
Let's consider 5 sentences $s_1, s_2, s_3, s_4, and\ s_5$with corresponding categories, where each pre-processed text word was considered inset of tokens(T). Let$\{T = T_1, T_2, T_3, T_4, T_5\}ands_j: j = 1,2,3,\ldots|S|$. In addition, the subset of token s_j is $s_j \subseteq T$. The sentences and tokens in $D_{pre-process}$taken for illustration are shown in Table III.
Table III: Sentence and Tokens of Input Data: An illustration

Sentence	Tokens				
S1	T_1	T_2	T_3		
S2	T_1	T_2	T_3	T_4	T_5
S3	T_1	T_2	T_3	T_4	
S4	T_1	T_2	T_3	T_4	
S5	T_1	T_2	T_3	T_4	T_5

Step 2: For each sentence, the category_id score is computed based on the frequency of the occurrence of tokens in sentences.
For illustration the computation of the category_id score, let's consider for$s_j = \{T_1, T_2, T_3\} \subseteq T$.

***Id_1*-Calculation:** The category_id score for Id_1can be computed as per Eq. (1), where, $F_T(k)$ is the frequency of token T in$s_j: j = 1,2,3,\ldots|S|$.

$$Id_{1j} = \frac{1}{|s_j|}\sum_{T \in s_j}\frac{1}{F_T(k)} \quad (1)$$

For illustration, while computing the occurrence of a token in sentence 1, the computation can be made as per Eq. (2). Here, tokens were given by T_{k1}, T_{k2}, T_{k3}appears in all 5 sentences, such that,
$F_{Tk}(1)$ =5,$F_{Tk}(2)$ =4 and $F_{Tk}(3)$ =3

$$Id_{11} = \frac{1}{3}\left\{\begin{matrix}\frac{1}{5} + & \frac{1}{4} + & \frac{1}{3} \\ \uparrow & \uparrow & \uparrow \\ T_{k1} & T_{k2} & T_{k3}\end{matrix}\right\} \quad (2)$$

***Id_2*-Calculation:** The Category_id score for Id_2can be computed by using Eq. (3).

$$Id_{2j} = \frac{|s_j|}{\sum_{T_k \in S_j} \frac{1}{F_{Tk}(k)}} \quad (3)$$

For illustration, the Category_id score for the 2nd id for corresponding to 1st sentence can be modelled as per Eq. (4).

$$Id_{21} = \frac{3}{5+4+3} = \frac{3}{12} = \frac{1}{4} \quad (4)$$

Each token has frequency i.eT1 has 5, T2 has 4 and T3has 3 frequency $F_T(k)$for sentence 1

Id_3Calculation: The Category_id score for $\boldsymbol{Id_3}$can be computed by using Eq. (5).

$$Id_{3j} = \frac{1}{|s_j|} \sum_{T \in S_j} \frac{F_T(k)}{|T|} \quad (5)$$

Eq.(5) can be written as per Eq. (6).

$$Id_{3j} = \frac{1}{|s_j||T|} \sum_{T \in S_j} F_T(k) \quad (6)$$

Thus, for 3rd id score for same the category corresponding to 1st sentence can be expressed as per Eq.(6). Here, three tokens appear for 5 times in s_1.

$$Id_{31} = \frac{1}{3\times5}(5+4+3) = \frac{12}{15} = \frac{4}{5} \quad (7)$$

(d) The selected sentence $s_j$for T depends on the distribution of Vocabulary words $\boldsymbol{\beta}$.

Then, the identified category based on the category_id score is denoted as $D_{category}$, which is subjected to mapping to BERT model.

Hybrid summarization using BERT fine tune model BERT based Extractive summary Model

The sentences $\boldsymbol{s_j}$ in the identified category $D_{category}$ are mapped to an "n-dimensional vector of real numbers". The BERT Model is designed to "map the sentences to capture the context". The BERT was designed with the intention of training the deep bidirectional representations transformers from the unlabelled text by conditioning together on the right and left contexts. Typically, the BERT Model encloses three major parts: Input layer, BERT encoder and output layer. For illustration: if there are two sentences, 'we went to the river bank' and 'I need to go to the bank to make a deposit'. In both these sentences, the token 'bank' is common, when are mapped under similar category they becomes meaningless, therefore the nature of the word 'bank' need to be identified to make it domain specific or category with higher precision. The BERT considers both the right and left context before mapping the words of a sentence under a category. Thus, the architecture is denoted as BERT extractive summary and it is shown in Fig.1

Input layer: The input sequence is constructed for the model by means of building "auxiliary sentence and the task" is turned into sentence-pair. An input sequence represents the sequence of texts or simpler texts in a token sequence, among which the first token is [CLS] and it contains the "special classification embedding", while the other is the special token [SEP] and it is utilized for separating segments or denoting the end of the sequence.

Adaptive BERT encoder: It is a "multi-layer bidirectional ttransformer encoder" that is modelled on the basis of the original implementations. It encapsulates "12 layers (Transformer blocks) and 12 self-attention heads". The extracted $D_{category} = \{Id_1, Id_2, Id_3, T_1, T_2, .. T_N\}$ from dataset given as input given to the encoder.

Output layer: In this proposed approach a fine tune model is required to generate informative summary. Therefore RNN-LSTM-CNN deep learning model is adapted

to fine tune BERT model by associating a relation between the original text Data, Label(extractive summary) and Category as input for further process. Here this model replaces soft max classifier in the output layer of BERT to train the vocabulary in the dictionary for abstractive summary using deep learning model and to solve OOV problem. The weight and bias factors being stable in the trained model (BERT); it is bit complex to process the natural languages of any data scale with higher accuracy. To make the categorization more precise, the category ids are adjusted with tokens in BERT model.

Fine Tune BERT with Modified Output Layer

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 the hidden Markov model predict the work by 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 1: 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)
STEP 4.2: 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:

$H\rightarrow tH\leftarrow t=\phi(XtW(f)xh+H\rightarrow t1W(f)hh+b(f)h),=\phi(XtW(b)xh+H\leftarrow t+1W(b)hh+b(b)h)$

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 $W(b)hh \in Rh \times h$, biases $b(f)h \in R1 \times h$ and $b(b)h \in R1 \times h$ for both forward and backward direction are the model parameters.

Further concatenating forward and backward layers using hidden layers $H\rightarrow t$ and Ht to obtain the hidden state of the word and label relation. Finally, $Ht \in Rn \times 2h$ 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 $0t \in Rn \times q$ (number of outputs: q): $0t = HtWhq + bq$.

Here, the weight matrix $Whq \in R2h \times q$ and the bias $bq \in R1 \times q$ are the model parameters of the output layer. The numbers of hidden layers will be different in both directions.

Algorithm 2: Hybrid Summarization Model

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

STEP1:Apply BERT based Extractive summarization
STEP2: Apply algorithm 1 by Fine tuning the Label generated by 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 IV
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. 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.

B-HEATS Frame work

An extraction of pre-labelled categories [26] are obtained from DUC 2002, DUC 2003 and DUC 2004 datasets.

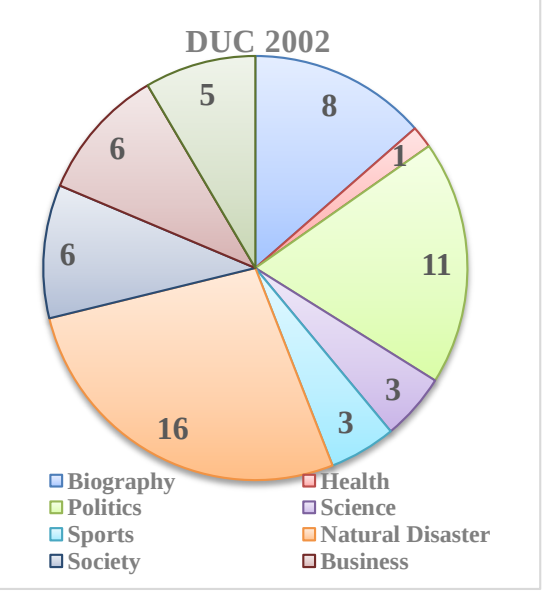

Fig 2: Category distribution on DUC2002 dataset

From the available source documents categories were identified and labelled accordingly. The traced categories from the datasets mentioned above are

'Biography, Health, Politics, Science, Sports, Natural Disaster, Society, Business and Culture. These pre-labelled categories and the corresponding documents count vary in each dataset were illustrated in Fig 2. For example, in DUC 2002 dataset, eight merged documents belong to Biography and 11 documents belong to politics, similarly for other categories respectively. Dataset was trained using BERT after pre-processing. The variation of documents from pre-labelled categories on DUC2003 and DUC2004 are presented in Fig 3 and Fig 4.

The fixed number of categories will be known in advance for MDS articles. In the training phase, the documents represent the sentences and the words associated with each category are considered as tokens. Then, the sentences and their associated tokens were mapped to an “n-dimensional vector of real numbers” under a specific category. The extractive summary is generated from mapping the “sentences to BERT to check context”, which are taken as a label for further Abstractive process.

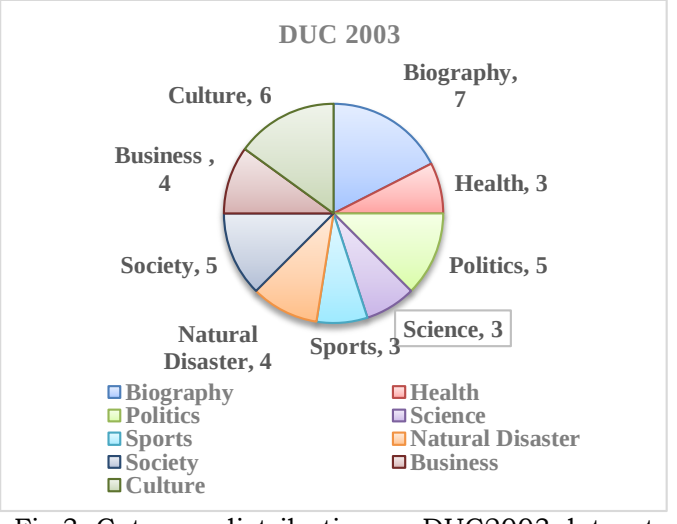

Fig 3: Category distribution on DUC2003 dataset

BERT is the first fine-tuning-based representation model that achieves state-of-the-art performance on a large suite of sentence-level and token-level tasks, outperforming much task-specific architecture. *The* BERT Model is fine-tuned with another deep learning architecture for the Final summary as pretrained models are unavailable for these un-labeled Datasets. The mapped sentence generated for each of the tokens, and sentences are captured in how they appear. Then summarizer groups the similar sentences into a single sentence by eliminating redundant words. Then the single sentence and context-based sentences are concatenated to generate a summary.

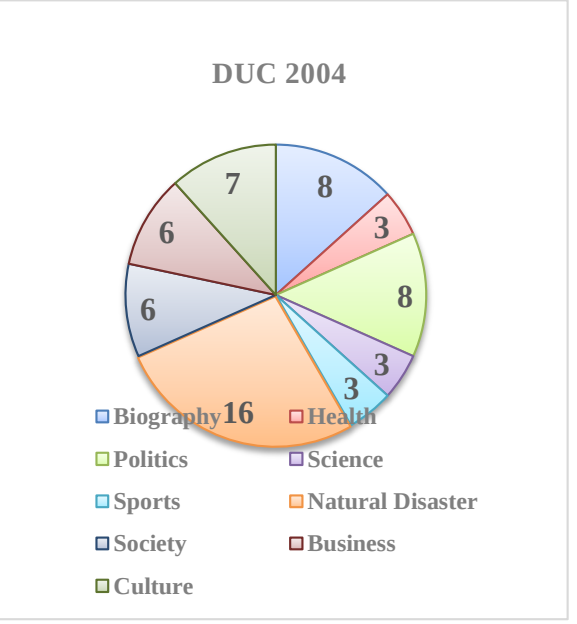

Fig 4: Category distribution on DUC 2004 dataset

For Example: When a document related to a category-politics is given as an input to the BERT model, the resultant tokens of each sentence s1 are {T1, T2, T3} for sentence one and {T1, T2, T3, T4, T5} for sentence s_2. The same process proceeds for the next sentences also. Thus, all the sets of tokens concerning sentences were assigned to the same category during the training phase.

During the testing phase, sentences and their associated tokens for test documents were subjected to BERT verification which identifies the category for each sentence. If most of the words belong to different categories, then sentences might vary according to that. On this condition, the example may be represented as s_1, s_2, s_5 of a document belongs to Politics and s_3, s_4 relates to Science and Biography respectively. In this approach, the maximum count of category words obtained by the sentences will be considered a category of the document$D_{category}$.

This approach helps to generate abstractive summary and the document belongs to a category that yields a qualitative word replacement.Category based sentence and document classification creates a dictionary of vocabulary belong to the category. This helps in solving Out of Vocabulary (OOV) problem and eliminates the sentences with that are not belonging to category of the Document. At the end, the final summary is generated by selecting $D_{summary}$informative sentences from all the clusters.

Results and Discussion

The proposed Novel Hybrid Automatic text summarization B-HEATS framework using deep learning model experimental results are evaluated on Benchmark datasets: “DUC 2002, DUC 2003 and DUC 2004”. The dataset description is tabulated in Table IV. The proposed B-HEATS model is compared over the existing

models like BERT [1], Bi-LSTM [8], and QODE [18] in terms of ROUGE scores ROUGE-1 (R1) ,ROUGE (R2), ROUGE-3 and ROUGE-L(RL) as well.The DUC Datasets consists of all documents belong to news domain with a diverse category documents in one cluster. Hence in DUC documents the 11 categories are extracted manually such as Biography, Culture, Business, Health, Politics, Law, Society, Natural Disaster, Science, Sports and International. The proposed summarization system compiles the documents in a cluster into a single merged document.

Table I
Description of DUC Datasets [26]

Dataset	Category	Cluster	Doc	Ref	Limitation
DUC 2002 [33]	Biography Culture Business Health Politics	59	567	116	100 words
DUC 2003	Law Society Natural- Disaster	40	477	117	100 words
DUC 2004	Science Sports International	60	500	200	100 ds

Category Selection using Fine tune BERT model

The proposed approach provides the category of the text data before classification which yields the better rate of summary compared with BERT and other encoding schemes along with N-Grams model. The data converted into a 3 column dictionary database (Category, Label and tokens) in B-HEATS model with the help of tokenized category. In fig 5, classification of categories in a cluster is depicted with the help of confusion bar chart. The x-axis in the graph shows extracted categories from DUC datasets. The categories represent the tested documents and the graph represents the perfect and imperfect categories identified i.e., true positives, false positives, false negatives and true negatives. For Sample only few categories i.e. Biography, Politics, Natural disaster and society are presented in a cluster because of maximum document categories. Total 8 documents are given for testing, out of 8 documents 5 shows exact category as biography, other 2 documents in politics and 1 detected as society.

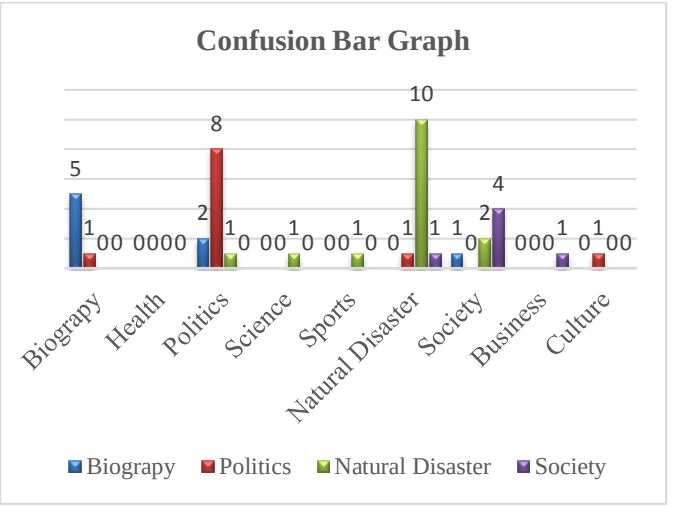

Fig 5 Category distribution in clusters

Evaluation Metrics

ROUGE scores is an evaluation metric to measure quality of summary based on number of overlapping units such as N-grams word pairs and word sequences between the system summary and reference summary. ROUGE values achieve high scores as system results matches with in human or reference summaries.
The Recall-Oriented Understudy for Gist Evaluation (ROUGE) scoring algorithm calculates the similarity between a system summary document and a collection of reference summary documents. Use the ROUGE score to evaluate the quality of document translation and summarization models.

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

Given an n-gram length n, the ROUGE-N metric between a system summary document and a single reference document is given by

$$ROUGE - N_{single}(systemsummary, reference) = \frac{\sum_{ri \in reference} \sum_{n-gramri} count(n-gram, system\ summary)}{\sum_{ri \in reference} numNgrams(r_i)} (8)$$

Where 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.

Longest Common Subsequence (ROUGE-L)

Given a sentence d=[w_1,...,w_m] and a sentence s, where the elements s_i correspond to words, the subsequence [w_{i1},...,w_{ik}] is a common subsequence of d and s if $w_{i'_j} \in \{s_1,...,s_n\}$ 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. The subsequence $[w_{i1},\ldots,w_{ik}]$ is a longest common subsequence (LCS) if the subsequence length k is maximal.

Given a system summary document and a single reference document the union of the longest common subsequences is given by

$$LCS_{i,j}(systemsummay, reference) = \bigcup_{ri \in reference} \{W | W \in LCS(systemsummary, r_i)\} \quad (9)$$

Where LCS (system summary, r_i) is the set of longest common subsequences in the system summary document and the sentence r_i from a reference document. Here system summary is denoted as SS in next portion of the paper.

The ROUGE-L metric is given by

$$R_{score} = \frac{\sum_{ri \in refernce} |SS, ri|}{numWords(reference)} \quad (10)$$

$$R_L(SS, reference) = \frac{(1+\beta^2) R_{score}(SS, reference) P_{score}(SS.reference)}{R_{score}(SS, reference) + \beta^2 P_{score}(SS, reference)} \quad (11)$$

The ROUGE-L metric calculates between a system summary and a reference summary. The proposed framework uses Benchmark datasets DUC 2002, DUC 2003 and DUC 2004 datasets. The description of datasets and average values for each dataset are given in the Table V. DUC 2002, DUC 2003 and DUC 2004 is comprised of 59,30 and 50 different clusters with diverse categories.

Table III
Dataset Description

Dataset Attributes	DUC 02	DUC 03	DUC 04
No. of Clusters	59	30	50
Min. No. of documents in each cluster	8	8	8
Average number of sentences per document	26	24	27
Word count in a document	250	250	250

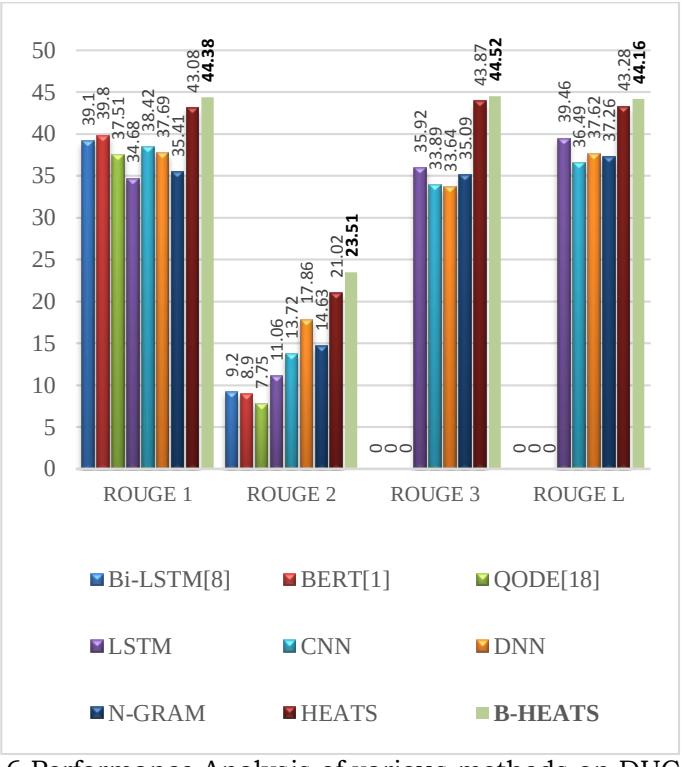

Fig 6 Performance Analysis of various methods on DUC-02

In Fig 6, the performance of the proposed B-HEATS framework compared with existing and generated models. In this paper the proposed system attains high relevant scores of original text in terms of ROGUE-1, ROGUE-2, ROGUE-3 and ROUGE-L. When compared to existing models the proposed framework B-HEATS Framework varies from 1.07 to 10.86 score level compared with 8 several text summarization models.

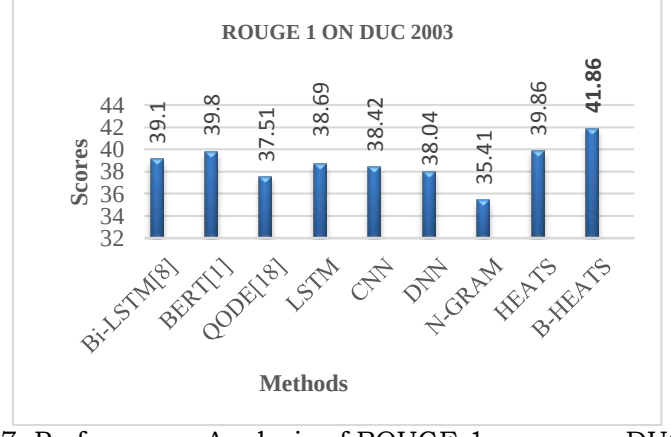

Fig 7. Performance Analysis of ROUGE-1 scores on DUC-03

In Fig 7, the performance of proposed B-HEATS framework is compared with existing and generated models. In this paper, B-HEATS achieves more relevant scores of original text in terms of ROGUE-1 i.e. 41.86 on DUC 2003 dataset which means unigrams similarity with original text. When compared to existing models, B-HEATS model varies from 2 to 2.7 in ROUGE scores when compared with eight summarization methods.

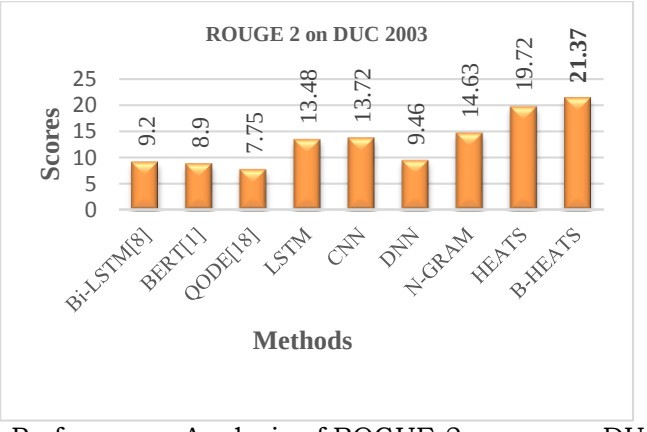

Fig 8: Performance Analysis of ROGUE-2 scores on DUC 03

In Fig 8, the B-HEATS Framework performance compared to existing and generated models on DUC 2003. The proposed B-HEATS achieves more relevant scores of original text in terms of ROUGE-2 i.e. 21.37 which indicates bi-grams similarity of summary with original text. When compared to existing models the, proposed one B-HEATS approach was varied from 2.65 to 12.17 score level compared with 8 different algorithms.

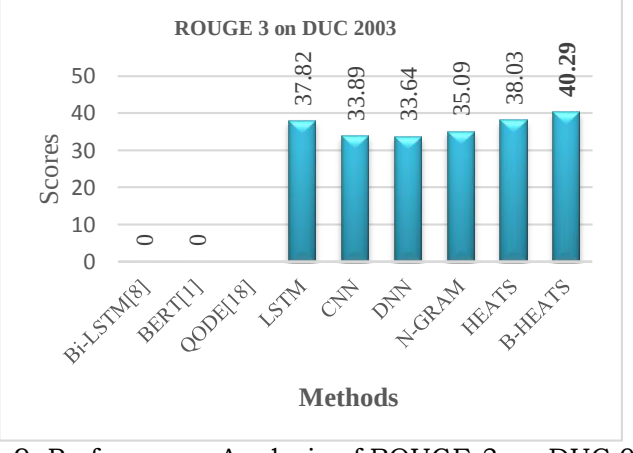

Fig 9: Performance Analysis of ROUGE-3 on DUC 03

In fig 9, the performance of proposed system is compared with existing and generated models on DUC 2003 Dataset. In this paper, B-HEATS framework attains more relevant scores of original text in terms of ROGUE-3 i.e. 40.29 .When

compared to existing models the proposed B-HEATS approach varied from 2.26 to 3.73 score level compared with 8 different Methods.

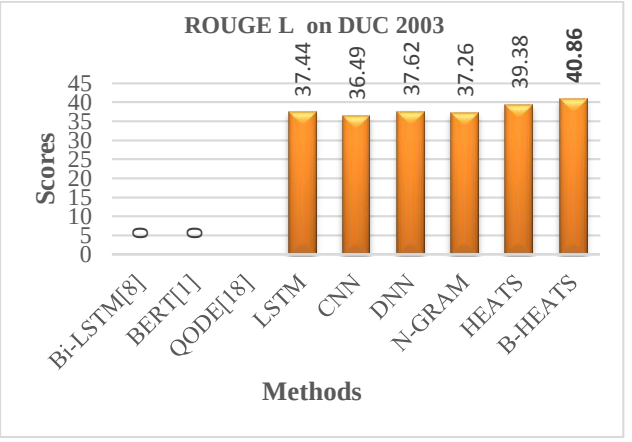

Fig 10: Performance Analysis of B-HEATS on DUC 03

In Fig 10, DUC 2003 based performance was compared with existing and generated models. In this paper, B-HEATS as proposing model provide more relevant scores of original text in terms of ROGUE-L i.e. 40.86. When compared to existing models, proposed B-HEATS approach varied from 0.48 to 3.42 score level compared with 8 different algorithms.

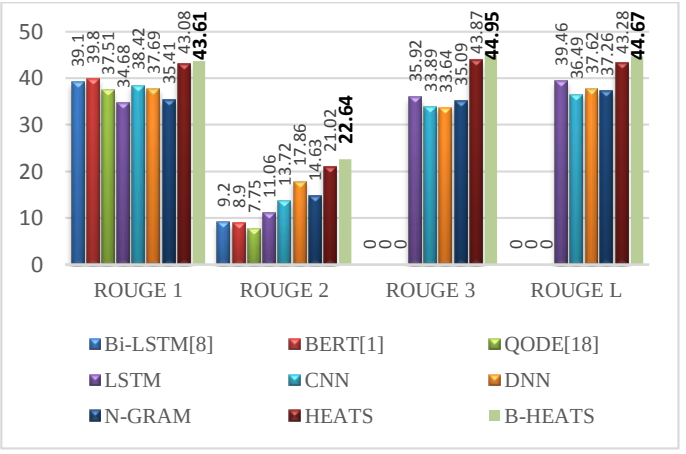

Fig 11. Performance Analysis of various methods on DUC 04

In Fig 11, DUC 2004 based performance was compared with existing and generated models. In this paper, B-HEATS as proposing model provides more relevant scores of original text in terms of ROGUE-1, ROGUE-2, ROGUE-3, and ROGUE-L. When compared existing models to proposed one B-HEATS approach and provide best aspect results i.e., R1 score 43.61, R2 score 22.64, R3 score 44.95 and RL score is 44.67 compared with 8 different algorithms. Qualitative measures like Redundancy, Informativeness and fluency are also calculated to evaluate summary quality.

Mathematically these were given as

$$Redundancy = Count\left(\frac{\sum(W_i(SS) \cap W_{i-1}(SS))}{Total\ words\ in\ SS}\right) (12)$$

Where $W_i$are words from System Summary(SS)

$$Informativeness = \frac{\sum(W_{oi}(SS) \cap W_i(SS))}{Total\ words\ in\ SS} \quad (13)$$

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

$$Fluency = \frac{\sum(W_{oi}(SS) \cap W_i(SS))}{Total\ words\ in\ Original\ text} (14)$$

Table IIII
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	12	78	83
B-HEATS (Proposed)	6	86	92

Table IVI
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
B-HEATS	8	82	96

In Table VII and Table VIII an abstract of 100 words were observed and measured for redundancy check, in-formativeness and fluency. In B-HEATS, the redundancy indicator measures whether the summary contains repeated information or not. On DUC 2003 it is showing only 6 and in DUC 2004 only 8 redundant words. The informativeness indicator can reflect whether the summary covers relevant points from the original merged documents. On DUC 2003 it is showing 86 and in DUC 2004 only 82 informative words, hence the summary is relevant. The fluency indicator emphases on whether the summary is well-formed and grammatical. On DUC 2003 it is showing 92 and in DUC 2004 only 96 words, hence the summary is readable. As the stop words were removed the transformed data is difficult to read therefore a sentence corrector was adopted later and these qualitative parameters were observed. The proposed scheme attains better results

than the existing ones, with a perfect read of 87 words. The results tabulated were averaged and the original varies in between 75 to 96.

Conclusion

In this research work, a novel Hybrid multi document text summarization using fine-tune BERT with category identification called B-HEATS framework was proposed. The given merged text documents are subjected to pre-processing, then identification of category is done based on category_id score for each category which is encoded in BERT. The extractive summary generated using BERT in this research work to map the sentences and fine tune using deep learning model.In B-HEATS framework**,** the output of BERT is fine-tuned using categorization and deep learning approach. The proposed automatic text summarization compared over the existing models in terms of evaluation metrics like ROUGE 1 (R1), ROUGE 2 (R2), ROUGE 3 (R3) and ROUGE L (RL) scores. The R1, R2, R3 and RL of the proposed work for DUC 2002 data is more than 0.32%, 0.44%, 0.25% and 0.2% better than the existing models like Bi-LSTM, BERT, QODE, LSTM respectively. The R1, R2, R3 and RL of the B-HEATS for DUC 2003 data is 0.6%, 0.46%, 0.43% and 0.56% better than the existing models like respectively. The R1, R2, R3 and RL of the proposed framework for DUC 2004 data is 0.74%, 0.46%, 0.75% and 0.63% better than the existing models like Bi-LSTM, BERT, QODE, LSTM respectively. Qualitative factors are also taken into consideration for summary quality. For future improvements, the model can incorporate optimization algorithms and can be extended to multilingual summaries also.

References

[1] Milad Moradi, Georg Dorffner, Matthias Samwald,"Deep contextualized embeddings for quantifying the informative content in biomedical text summarization",Computer Methods and Programs in Biomedicine, 2019

[2] Jinming Zhao, Ming Liu,Longxiang Gao,Yuan Jin,Lan Du and He Zhao,"SummPip: Unsupervised Multi-Document Summarization with Sentence Graph Compression",SIGIR,2020

[3] Akanksha Joshi, E. Fidalgo,E. Alegre,Laura Fernández-Robles,"SummCoder: An unsupervised framework for extractive text summarization based on deep auto-encoders",Expert Systems With Applications,vol.129,pp.200-215,2019

[4] Deepa Anand and Rupali Wagh,"Effective deep learning approaches for summarization of legal texts",Journal of King Saud University – Computer and Information Sciences,2019

[5] Qasem A. Al-Radaideh and Dareen Q.Bataineh, " A Hybrid approach for Arabic text summarization Using Domain Knowledge and Genetic algorithms", Cognitive Computation, March, 2018

[6] Shengli Song,Haitao Huang & Tongxiao Ruan ,"Abstractive text summarization using LSTM-CNN based deep learning",Multimedia Tools and Applications , vol.78,pp.857-875,2019

[7] Nabil Alami, Noureddine En-nahnahi, Said Alaoui Ouatik & Mohammed Meknassi ,"Using Unsupervised Deep Learning for Automatic Summarization of Arabic Documents",Arabian Journal for Science and Engineering , vol.43, pp.7803-7815,2018

[8] Minakshi Tomer & Manoj Kumar,"Improving Text Summarization using Ensembled Approach based on Fuzzy with LSTM",Arabian Journal for Science and Engineering,2020
[9] Zhenrong Deng, Fuxin Ma, Rushi Lan, Wenming Huang, Xiaonan Luo,"A Two-stage Chinese text summarization algorithm using keyword information and adversarial learning",Neurocomputing, in communication, 2020
[10] Asad Abdi, Siti Mariyam Shamsuddin, Shafaatunnur Hasan, Jalil Piran,"Machine learning-based multi-documents sentiment-oriented summarization using linguistic treatment",Expert Systems with Applications,2018
[11] Nabil Alami, Mohammed Meknassi, Noureddine En-nahnahi,"Enhancing unsupervised neural networks based text summarization with word embedding and ensemble learning",Expert Systems with Application,2019
[12] Arturo Curiel, Claudio Gutiérrez-Soto, José-Rafael Rojano-Cáceres,"An online multi-source summarization algorithm for text readability in topic-based search",Computer Speech & Language, in communication,2020
[13] X. Lin, M. Liu and J. Zhang, "A Top-Down Binary Hierarchical Topic Model for Biomedical Literature," in IEEE Access, vol. 8, pp. 59870-59882, 2020, doi: 10.1109/ACCESS.2020.2983265.
[14] Rupal Bhargava, Yashvardhan Sharma,"Deep Extractive Text Summarization",Procedia Computer Science,2020
[15] Shengluan Hou, Ruqian Lu,"Knowledge-guided unsupervised rhetorical parsing for text summarization",Information Systems,2020
[16] Rupal Bhargava, Gargi Sharma, Yashvardhan Sharma,"Deep Text Summarization using Generative Adversarial Networks in Indian Languages",Procedia Computer Science,2020
[17] Amy J. C. Trappey, Charles V. Trappey, Jheng-Long Wu, Jack W. C. Wang,"Intelligent compilation of patent summaries using machine learning and natural language processing techniques",Advanced Engineering Informatics,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] Farooq Zaman, Matthew Shardlow, Saeed-Ul Hassan, Naif Radi Aljohani, Raheel Nawaz ,"HTSS: A novel hybrid text summarisation and simplification architecture",Information Processing & Management,2020
[22] Adhika Pramita Widyassari, Supriadi Rustad, Guruh Fajar Shidik, Edi Noersasongko, De Rosal Ignatius Moses Setiadi,"Review of automatic text summarization techniques & methods",Journal of King Saud University - Computer and Information Sciences,2020
[23] Min Yang, Xintong Wang, Yao Lu, Jianming Lv, Chengming Li,"Plausibility-promoting generative adversarial network for abstractive text summarization with multi-task constraint",Information Sciences,2020

[24] Jiyuan Zheng, Zhou Zhao, Zehan Song, Min Yang, Xiaohui Yan,"Abstractive meeting summarization by hierarchical adaptive segmental network learning with multiple revising steps",Neurocomputing,2020
[25] Duy Duc An Bui, Guilherme Del Fiol, John F. Hurdle, Siddhartha Jonnalagadda,"Extractive text summarization system to aid data extraction from full text in systematic review development",Journal of Biomedical Informatics,2016.
[26] Cao, Ziqiang & Li, Wenjie & Li, Sujian & Wei, Furu, " Improving Multi-Document Summarization via Text Classification", 2016.
[27] Upadhyay, Abhishek, Javed Khan Ghazala, Balabantaray, Rakesh Chandra, Rautray Rasmita,'Multi-document Summarization Using Deep Learning', 'Intelligent and Cloud Computing', Springer, Year 2021.
[28] Rush, Alexander & Chopra, Sumit & Weston, Jason. 'A Neural Attention Model for Abstractive Sentence Summarization'. Comput. Sci. , Year 2015.
[29] Yuliska and T. Sakai, 'A Comparative Study of Deep Learning Approaches for Query-Focused Extractive Multi-Document Summarization', 2019 IEEE 2nd International Conference on Information and Computer Technologies (ICICT), Year 2019
[30] Ren P., Z. Chen, Z. Ren, F. Wei., L. Nie., J. Ma. and M.D. Ridjke, 'Sentence Relation for Extractive Summarization with DeepNeural Network'. ACM Transaction on Information System (TOIS), 2018, Volume 36 Issue 4, Article No. 39.
[31] D. Park, S. Kim, J. Lee, J. Choo, N. Diakopoulos and N. Elmqvist, "ConceptVector: Text Visual Analytics via Interactive Lexicon Building Using Word Embedding," in IEEE Transactions on Visualization and Computer Graphics, vol. 24, no. 1, pp. 361-370, Jan. 2018, doi: 10.1109/TVCG.2017.2744478.
[32] E. Yulianti, R. Chen, F. Scholer, W. B. Croft and M. Sanderson, "Document Summarization for Answering Non-Factoid Queries," in IEEE Transactions on Knowledge and Data Engineering, vol. 30, no. 1, pp. 15-28, 1 Jan. 2018, doi: 10.1109/TKDE.2017.2754373.