ISSN 2007-9737

Automatic Text Summarization Using Sequence to Sequence Model and Recurrent Neural Network

Pooja Gupta[1,2], Swati Nigam[1,2], Rajiv Singh[1,2,*]

[1] Department of Computer Science, Banasthali Vidyapith, Rajasthan, India

[2] Centre for Artificial Intelligence, Banasthali Vidyapith, Rajasthan, India

{poojagupta2291, swatinigam.au, jkrajivsingh}@gmail.com

Abstract. Presently, the Internet serves as a repository for an extensive variety of content, encompassing scholarly articles, current affairs, blog posts, and social media updates. This category of electronic data comprises a vast quantity of information that requires management, organisation, and storage. Text summarization is a procedure that reduces the size of substantial amounts of textual data to a more feasible format. The process of summarising English-language literature can be approached in a variety of ways. Conversely, a limited number of them find application in low-resource languages such as Hindi. For the purpose of text summarization in English and Hindi, we implemented a sequence-to-sequence (Seq2Seq) encoder decoder model with a recurrent neural network (RNN) and presented two methods: an extractive method and an abstractive method. By applying the suggested methodology, we successfully extracted the most critical sentences from the input documents through the use of extractive text summarization. Abstractive text summarization utilizes a natural language generation approach to produce a summary that maintains the integrity of the original content. The fundamental objective of this research is to generate abstractive and extractive summaries in both Hindi and English for the identical datasets. We compiled the summaries using four datasets, two of which were the CNN News and BBC News datasets and were utilized for the English-Hindi parallel corpus. Conversely, for Hindi summaries only, the Indian language text corpus dataset and the Hindi text summarization corpus dataset are used. We exploit the ROUGE metric with specific parameters, including F-measure, precision, and recall, to assess the method. Experimental results are compared with the existing state-of-the-art methods.

Keywords. Abstractive summarization, extractive summarization, deep learning, Seq2Seq, recurrent neural network, ROUGE.

1 Introduction

The development of automatic text summarization for Hindi documents faces various problems, including a lack of big training datasets, a lack of parallel corpus, exceptionally long document summary pairs with corresponding parallel text, and so on. [1] To resolve this issue, we introduce a sequence-to-sequence encoder-decoder approach that generates new sentences based on the extraction of relevant lines from a news article.

To create the summaries, we used documents in Hindi and English. We use an RNN model to execute the sequence-to-sequence task in extractive summarization. Figures 1 and 8, respectively, depict the architectures of the proposed extractive and abstractive text summarizing techniques. In extractive text summarization, encoders read documents, whereas decoders extract sentences from the input document.

We use the extractive summaries as input to generate abstractive summaries, utilizing natural language generation techniques. We perform abstractive summarization on Hindi texts from two major news datasets, BBC News and CNN News, to train our model.

We tested using two different Hindi datasets: the Hindi text summarization corpus dataset and the Indian language text summarization corpus dataset. The BBC News and CNN News datasets are in English, so we translated them into Hindi using three open-source machine translators: Microsoft Bing, Google, and Systran. The

Seq2Seq model is the fundamental approach for generating abstractive summaries in English; however, it is more difficult in Hindi.

The encoder-decoder architecture solves the challenge of abstractive text summarization. This approach feeds an input document into an encoder, which then outputs an RNN model to generate the summary. We built the initial encoder-decoder model using the Seq2Seq approach [2]. Using a trained model, we implement a heuristic by re-ranking every sentence in a document according to its likelihood of being a summary sentence. We rank the results produced by N-gram language models using RNN.

Furthermore, we conducted a comparison between our results and those of other studies that used CNN and BBC datasets for processing. Although extracting summaries has been the focus of several studies, the current research contrasts with some of the most recent automatic text summary methods that have produced superior outcomes. We make comparisons with other current transformers, including PEGASUS [27], BERT [22], BART [26], and T5 [23].

The main contributions of this research are as under:

– We introduce a deep learning-based extractive and abstractive text summarization architecture for Hindi and English text documents.
– To summarize a news article, we employed a sequence-to-sequence encoder-decoder model using RNN. First, we extracted the key sentences for extractive summaries, and then we generated new sentences for abstractive summaries.
– Through experiments, we demonstrate that the proposed method outperforms other current methods and fundamental systems. The proposed method obtains a notable performance boost on the CNN News dataset, the BBC News dataset, the Hindi Text Summarization Corpus dataset, and the Indian language text summarization corpus dataset.

2 Literature Review

The use of extractive text summarization in deep neural network (DNN) information processing has gained popularity. These machine learning techniques integrate numerous nonlinear neural network (NN) layers. To function well, the DNN requires a large amount of training data.

For example, the development of DNNs such as recurrent neural networks (RNN) and convolutional neural networks (CNN) requires enormous datasets. Abstractive summary methods require NLP, machine learning, and deep learning methods [3, 8, 21] to generate and select meaning words to form new sentences.

Other recent abstractive methods are Pointer Generator [17], Pre-Training with Extracted Gap-Sentences for Abstractive Summarization (PEGASUS) [27], and T5 [23]. Nowadays, abstractive text summarization is based on a neural network that produces a neural sequence-to-sequence model. A hybrid pointer-generator network model uses this model on various datasets, including CNN and the Daily Mail.

The PEGASUS model is a newly proposed Google model for pre-training large corpora via a transformer-based encoder-decoder model [27]. There are two key graph-based methods that yield promising results for sentence ranking: TextRank [42] and LexRank [43].

For the Indian language, these methods are domain- and language-independent [44]. Both methods select words or phrases from an input document and position them as vertices in a weighted, undirected network. Edges are then drawn between sentence pairs according to how similar they are to each other.

The main difference between TextRank and LexRank is that they measure the similarity between two sentences. TextRank measures the similarity based on the similar words between two sentences, and LexRank measures the similarity by using cosine similarity.

The Google search engine uses the PageRank [45] algorithm in both methods to rank webpages and select important phrases through a random walk across a network.

Tables 1 and 2 describe previous work in extractive and abstractive text summarization based on datasets, approaches, methods, and challenges [3]. In the previous year's study, many researchers created summarizer models to generate understandable automatic text summaries. The majority of extractive text

Table 1. A comparison based on dataset, techniques, methods, and problem of extractive text summarization

Research	Year	Techniques	Methods	Challenges
Ren et al.,[8]	2016	ML	CNN	Sentence Scoring
Gulati et al.,[9]	2016	ML	Fuzzy logic	Extraction
Wu et al.,[10]	2017	ML	LDA	Topic modeling
Nalik et al., [5]	2017	Rule Based	Rule Based	Extraction
Fang et al., [7]	2018	ML	Co-rank	Sentence Ranking
Khan et al., [6]	2019	ML	TF-IDF and K-Means	Extraction
Lierde et al., [4]	2019	Statistic	Fuzzy hyper graph	Semantic
Alami et al., [11]	2021	ML	MMR	Semantic

Table 2. A comparison based on dataset, techniques, methods, and problem of abstractive text summarization

Research	Year	Dataset	Techniques	Methods	Challenges
Chopra et al., [14]	2016	DUC	DL	CNN	Long Sequences
Nallapati et al., [15]	2016	Gigaword, CNN/DM	DL	GRU-RNN	Unknown Words
Zeng et al., [16]	2016	Gigaword, DUC	DL	GRU LSTM	Large Vocabulary
See et al., [17]	2017	CNN/DM	DL	Bidirectional LSTM-RNN	Repeated Statements
Cao et al., [18]	2018	Gigaword	DL	Bidirectional GRU	Generating Summaries with Fake Facts
Sahoo et al., [13]	2018	DUC 2002	ML	Markov and SVM	Sentence Scoring
Azmi et al., [12]	2018	Arabic	Statistic	TF-IDF and NLP	Ambiguity
Zhang et al., [19]	2019	Gigaword, CNN/DM	DL	CNN	Sequential Nature of RNNs

summarization uses machine learning techniques. Machine learning is considered one of the most effective techniques in most studies due to its ability to generate modern parameters.

In the last ten years, researchers have employed word embedding, TF-IDF, SVM, K-means, Markov, and MMR (maximal marginal relevance) techniques. A lot of researchers will utilize both machine learning and deep learning to summarize abstract language.

Most of the research on abstractive text summarization has used deep learning algorithms like RNN, CNN, GRU, LSTM, bidirectional LSTM-RNN, and bidirectional GRU to create short summaries of short text documents on well-known datasets like XSUM, DUC, CNN/DM, and CNN. In the current study, generative tasks are utilized to produce abstractive summaries known as pre-trained language models (PTLMs).

These models have a comprehensive semantic and contextual set of features that serve to improve

Table 3. A comparison of abstractive text summarization related PTLMs

PTLMs	Year	Dataset	Methods
GPT [20]	2018	Book Corpus	Language modeling (LM)
ELMOs [21]	2018	One-Billion-Word	Bidirectional LM
BERT [22]	2019	English Wikipedia	Masked LM
T5 [23]	2019	C4	Masked Seq2Seq LM
UniLM [24]	2019	English Wikipedia etc.	Multi-task Seq2Seq masked LM
MASS [25]	2019	-	Masked Seq2Seq LM
BART [26]	2020	English Wikipedia etc.	Denoising auto-encoder
PEGASUS [27]	2020	C4, HugeNews	Masked LM
Prophetnet [28]	2020	English Wikipedia etc.	Future n-gram prediction
UniLMv2 [29]	2020	English Wikipedia etc.	Seq2Seq masked LM and Bidirectional LM
BigBird Pegasus [31]	2020	Big patents	Masked LM
Switch-C [30]	2021	Improved C4	Masked LM
Pegasus-X [32]	2022	XSUM, CNN/DM	Long input summarization
Switch Transformers (FFN) [33]	2022	Colossal Clean Crawled Corpus	T5-base and T5-large based

the readability and relevance of the generated summaries. Table 3 compares various existing and popular pre-trained language models based on datasets and methodologies, such as ELMos, BERT, T5, BART, and PEGASUS.

The results of text summaries, whether generated by machines or by humans, must be evaluated. However, the lack of a common evaluation criteria and the widespread usage of different criteria have made it challenging to evaluate text summaries.

For text summarization, two evaluation parameters are used: automatic evaluation and human evaluation. In automatic evaluation, the system's performance is measured by using the very popular metrics of ATS, i.e., ROUGE [40].

In human evaluation of text summarization, the human judgements of different quality metrics such as readability, structure, and coherence, grammatically, referential clarity, content coverage, conciseness, and non-redundancy were computed.

3 Proposed Work

The proposed extractive approach acquires input text documents and applies text pre-processing to these texts. Once pre-processing is done, the most important features from the sentences, such as word embedding, TF-IDF, and bag-of-words, will be extracted as part of the feature extraction procedure. The datasets are trained using the

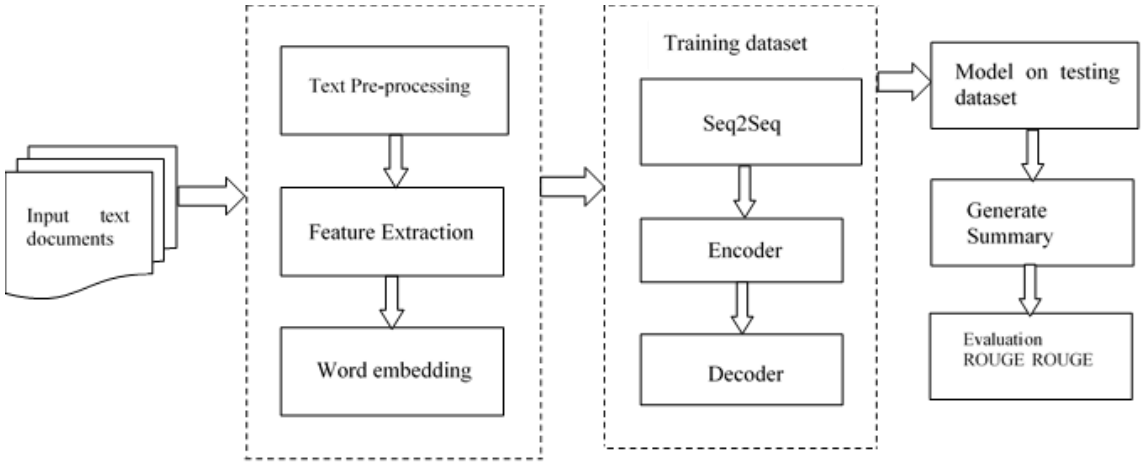

Fig. 1. Proposed extractive text summarization

Seq2Seq encoder-decoder model after word embedding. Figure 1 illustrates the proposed extractive text summarization approach and Figure 8 depicts the abstractive model using RNN.

3.1 Dataset Collection, Dataset Description and Cleaning of Corpus

This section defines the data used in this research, including articles from newspapers and websites in Hindi and English that were used to create extractive summaries. These articles were gathered from a variety of state-of the-art datasets, including BBC News[1], CNN/Daily-Mail News[2,] in English language.

Three open-source translators such as Microsoft Bing translator[3], Google translator[4], and Systran translator[5] [34] which are all freely available online, translated them into Hindi. We have perfromed on all three translators, but we have only take the Google translated sentences for generating the extractive Hindi summaries. Beacause of high performance, the Google translator gives the highest results in all evaluation metrics [35].

Hindi text short summarization corpus[6] and Indian language text corpus datasets[7] are loaded from the kaggle; it contains only Hindi text documents. Figure 2 depicts the BBC news dataset, which contains the following subfields: business, politics, entertainment, sports, and technology. The BBC news dataset contains 2225 entries over five fields.

Since the lengths of the documents are set at roughly ten sentences each, the summary length that results comprises three to four sentences chosen from the documents that received the highest score. The CNN dataset contains 500 news stories, with 400 utilized for training and 100 for testing. The document's summary and length are aligned with the BBC dataset.

Only Hindi text documents are included in the Indian language news text summarization corpus dataset and the Hindi text short summarization corpus dataset, which are loaded from Kaggle. Since the document lengths for the Indian language text corpus (ILSUM) and Hindi text brief summarization corpus datasets have been determined at about 20 sentences each, the final summary length contains 10 sentences that were

[1] https://www.kaggle.com/pariza/bbc-news-summary
[2] https://www.tensorflow.org/datasets/catalog/cnn_dailymail
3 https://www.microsofttranslator.com
4 https://translate.goolge.com
5 https://www.systran.net/en/translate/

[6] https://www.kaggle.com/datasets/disisbig/hindi-text-short-summarization-corpus
[7] https://www.kaggle.com/datasets/deekoul/isndian-language-summarization

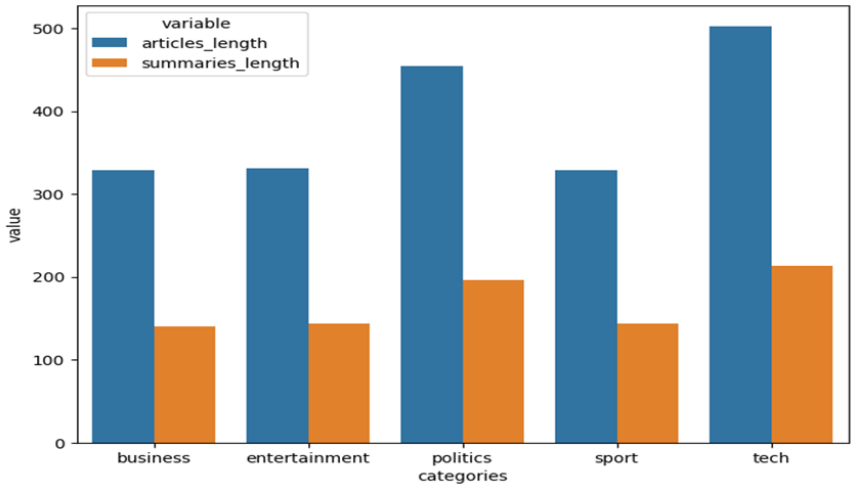

Fig. 2. Pictorial representation for length of the articles and summaries for BBC English dataset with individual categories

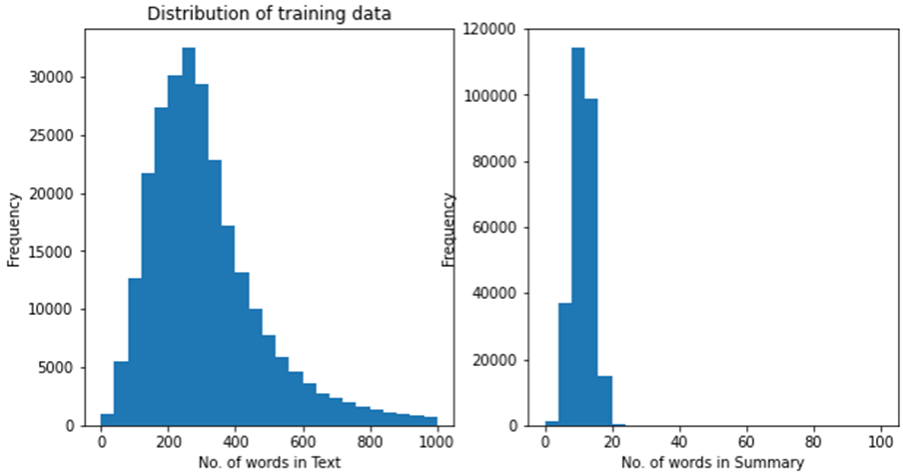

Fig. 3. Distribution of training data and number of words in summary for Hindi text summarization dataset

chosen from the documents that received the highest score.

The previous explanations outline how to use the Seq2Seq model to extract the key sentences for extractive text summarization. The abstractive text summarization's parameters and their values are listed in Table 4. The Seq2Seq model was run using the Google Collaboratory. The preceding datasets were separated into training and testing sets.

The proposed approach was tested with 10% of the data, while the remaining 90% was used for training. A tensor flow CPU version was used to construct our model. We propose an abstractive text summarizer for both English and Hindi text documents. There are many effective summarizers available in English. However, we have made a proposal to create an improved method for the same datasets in both languages.

We employed the encoder-decoder approach, with three layers in the encoder and three layers in

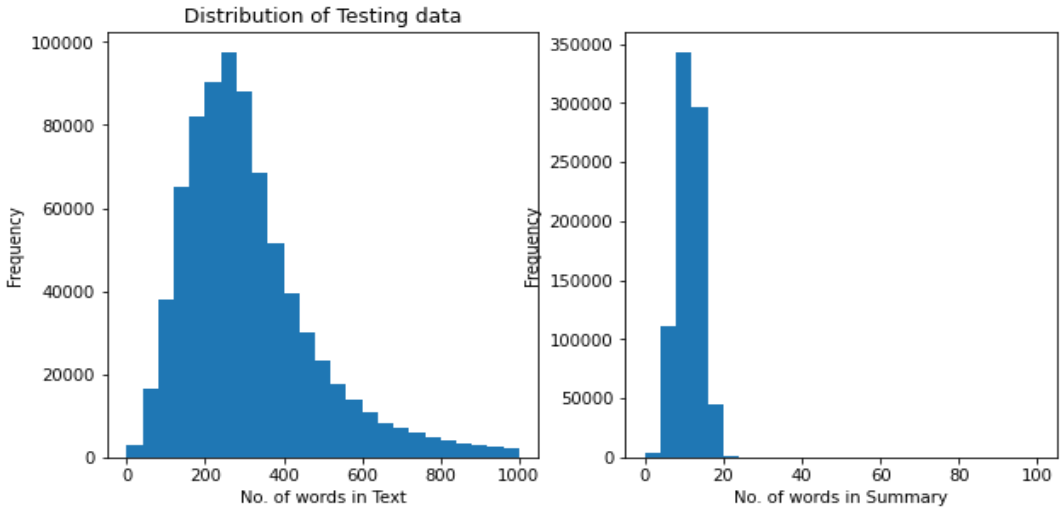

Fig. 4. Distribution of testing data and number of words in summary for Hindi text summarization dataset

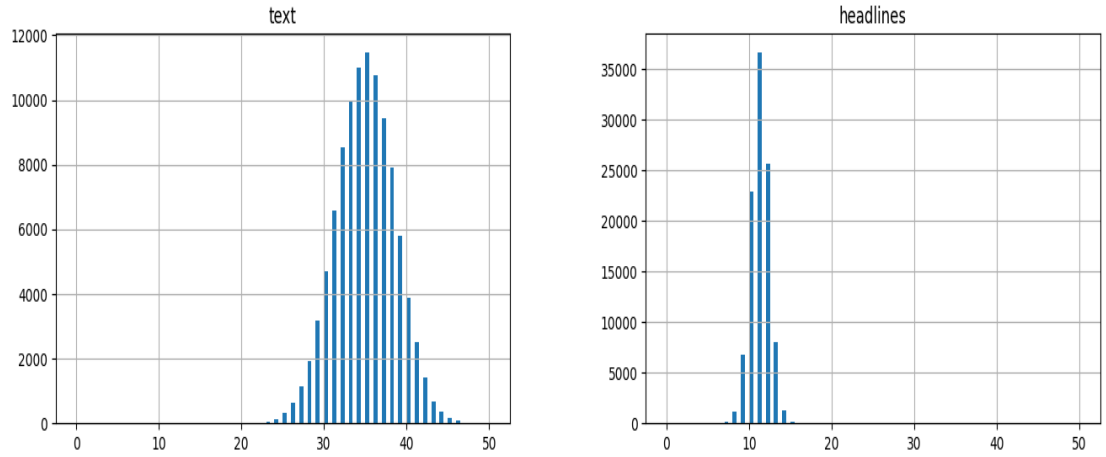

Fig. 5. Text summary plotting of Hindi text summarization dataset

the decoder, to train the proposed model. Word vector size and hidden state size are both 256 for training purposes. We have trained news datasets with appropriate articles and headlines by utilizing these factors.

Testing, validation, and training are all done with the datasets. The distributions of training and testing statistics for the Hindi text summarization dataset are displayed in Figures 3-4. For the Hindi text summarization dataset and the BBC News dataset, the text and generated summary are displayed in Figures 5 and 6, respectively.

3.2 Text Pre-Processing

In order to generate the summary, firstly we import an English input document from the Kaggle dataset and a Hindi document from the translated text files. As the result of different text being organized in unstructured forms on the internet, text pre-processing, which is necessary in many NLP applications, can be performed after accepting an input document.

Given that it contains noise in many forms, such as stop words, punctuation marks, emotions, and

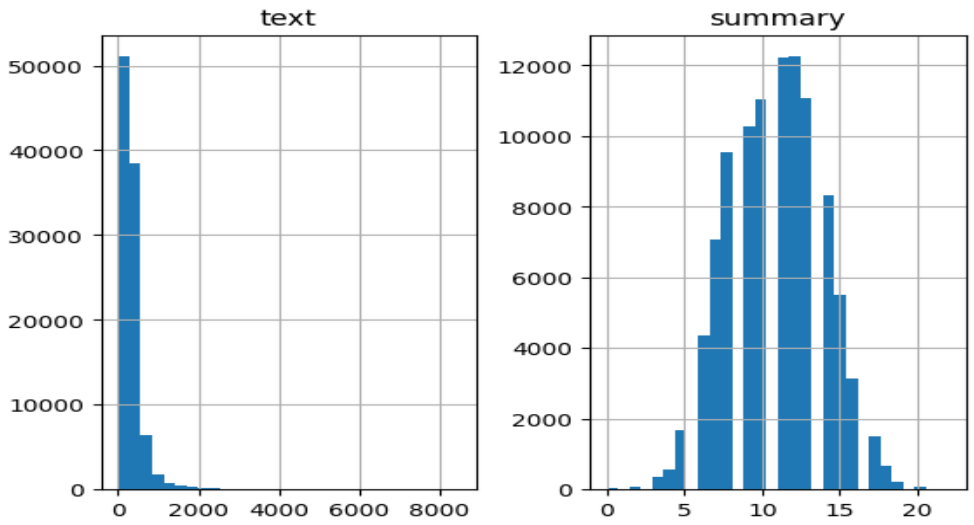

Fig. 6. Text summary plotting of BBC News dataset

distinct text instances, handling it in a language with limited resources like Hindi is an extremely difficult challenge. As a result, text pre-processing is necessary to make the text corpus cleaner.

Text pre-processing will be carried out in multiple steps, including tokenizing sentences, tokenizing words, removing punctuation, eliminating stop words, stemming, and lemmatization, as demonstrated in Figure 7. Sentences are divided into separate sentences and saved with their respective sentence positions during sentence tokenization.

Word tokenization divides the split sentences into individual words. Punctuation marks and stop words from the extracted words from the input sentences will be eliminated. The NLTK stop words list[8] was utilized for both Hindi and English text.

After stop word elimination, supervised statistical POS tagging will be used to assign the POS for each extracted word [36]. After POS tagging, stemming and lemmatization will be performed.

This work will be accomplished by using a lightweight stemmer, morphological analyzer, and rule-based stemmer [37]. Throughout the pre-processing stage, we utilized pre-trained models, including T5, BART, BERT, and PEGASUS, after the tokenization of words and sentences. The purpose of this is to make sure that the split text uses the same vocabulary from pre-training and correlates in the same way with the pre-trained model's corpus.

After finishing the pre-processing stage, sentence features can be extracted and the sentence score calculated. The sentences were analysed to determine sentence position, sentence length, TF-IDF, i.e., term frequency-inverse document frequency score, bi-gram and tri-gram scores, scores, and other metrics.

When these features are calculated, they produce a sentence matrix. These feature values will be sent to the training model. To improve our model, we applied a pre-trained word to a vector file named GloVe (Global Vectors for Word Representation).

3.3 Seq2Seq Model

The Seq2Seq Model's encoder layer will receive and handle these feature values. The Seq2Seq model's architecture is made up of two main components: the encoder and the decoder, both of which are RNN [38].

[8] https://www.nltk.org/nltk_data/

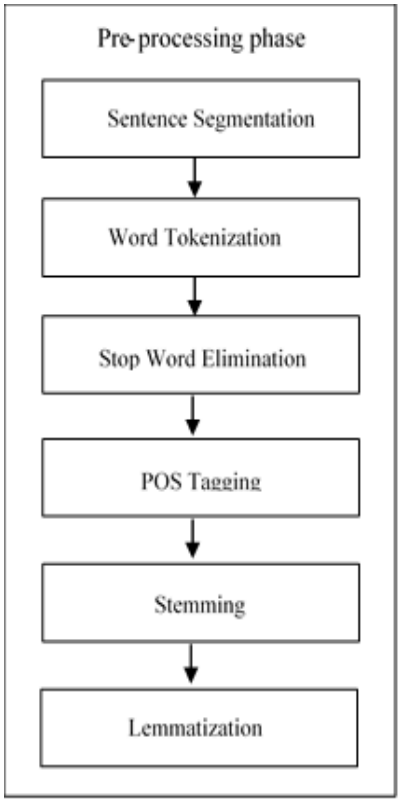

Fig. 7. Pre-processing phase

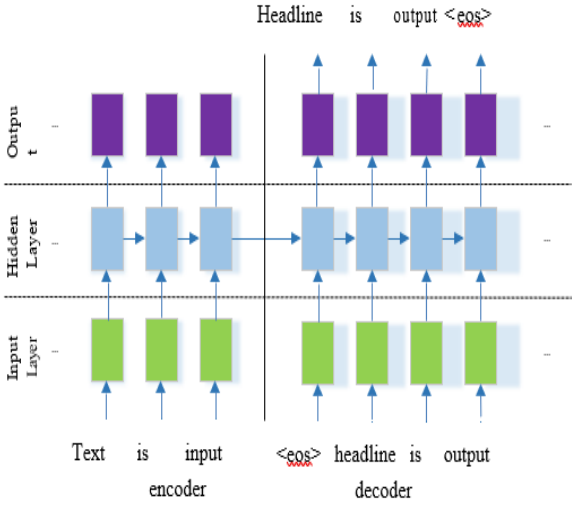

Fig. 8. Sequence-to-Sequence model with RNN for abstractive summarization

The encoder receives the text of the news article one word at a time. Each word is first transformed into a distributed representation by passing through an embedding layer.

A multi-layer neural network is then used to integrate this dispersed representation, which includes either all 0s for the first word in the text or the hidden layers formed after feeding in the prior word, as shown in Figure 8.

The decoder receives the final word of the input text and uses the newly generated hidden layers as input. An embedding layer is employed to convert the input EOS, i.e. end-of-sequence symbol back into a distributed representation. The decoder then generates text summaries for each word in the headline using a *SoftMax* layer and the attention mechanism described in the following section, before terminating with an end-of-sequence symbol.

Each word is formed, and the exact same word is used as the input for the next word. Some new special tokens, such as <UNK> and <EOS> have been introduced to the lexicon. Due to limited vocabulary few words are still in use. UNK defines the token takes the place of those words [39].

The end of the sequence, which the EOS token contains, signals the encoder when it receives input. *SoftMax* regression and multiclass classification both are used for calculating the loss of categorical cross entropy. We have calculated cross entropy loss by equation 1, where Y is the real value and K is the total number of classes in the dataset:

$$Loss = \sum_{j=1}^{K} Y_j \log(Y_j). \quad (1)$$

3.4 Summary Generation

We generate a summary after implementing the sequence-to-sequence model. We thus execute the model in the deep learning framework in order to generate the summaries. About 10 minutes are required for the Google Colab GPU accelerator to train the text documents. In order to train the model, we set a minimum batch size because training stops if the batch size is too large. Our model is trained over two epochs. Using our model, we have produced a summary after the training phase has been completed.

Table 4. Parameters and their values for abstractive text summarization

Parameters	Value
Language	English, Hindi
Input description length	100 words (English) 50 words (Hindi)
Output summary length	30 words (English) 15 words (Hindi)
Learning rate	0.01
Batch Size	8
Epochs	2
Uniform distribution	from {-0.1,0.1}

Table 5. Compare ROUGE scores between proposed and other SOTA approaches on BBC News and CNN News datasets for extractive text summarization

	BBC News dataset			CNN News dataset		
Approach	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L
TextRank	0.86	0.83	0.81	0.56	0.42	0.49
LexRank	0.78	0.73	0.75	0.45	0.3	0.39
Lead	0.77	0.73	0.74	0.61	0.51	0.52
Luhn	0.8	0.76	0.77	0.5	0.36	0.42
LSA	0.86	0.83	0.82	0.5	0.34	0.43
SumBasic	0.7	0.62	0.68	0.48	0.32	0.38
Proposed	**0.89**	**0.84**	**0.82**	**0.65**	**0.56**	**0.58**

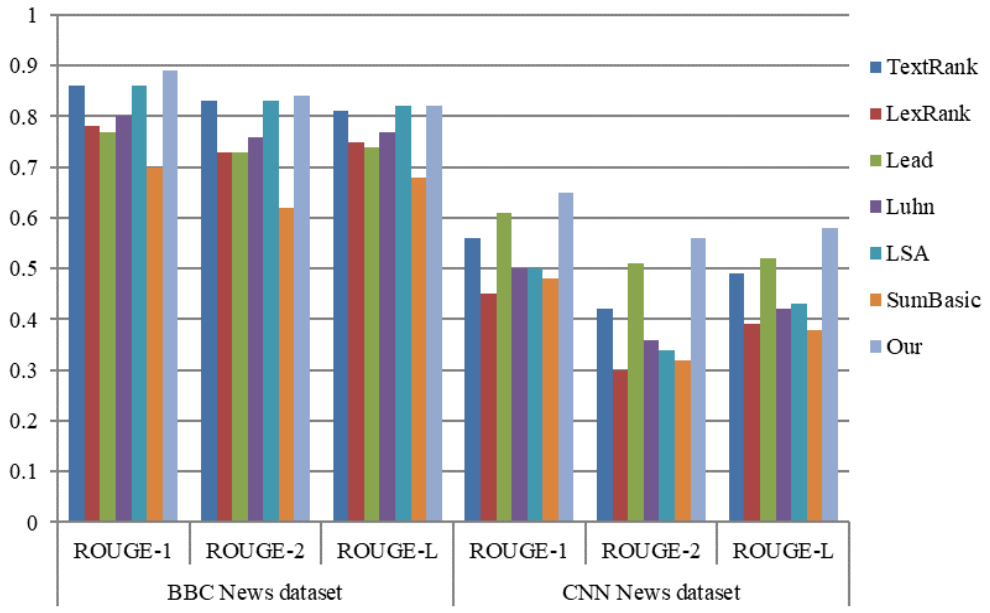

Fig. 9. ROUGE based comparison of proposed and other baseline approaches on BBC News and CNN News dataset for extractive text summarization

Table 6. Compare ROUGE scores between proposed and other SOTA approaches on Hindi text summarization corpus and ILSUM dataset for extractive text summarization

	Hindi Text summarization corpus dataset			**Indian language text summarization dataset**		
Approach	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L
TextRank	0.71	0.63	0.69	0.57	0.29	0.52
LexRank	0.63	0.53	0.63	0.52	0.31	0.47
Lead	0.62	0.53	0.62	0.53	0.31	0.5
Luhn	0.65	0.56	0.65	0.52	0.31	0.46
LSA	0.71	0.63	0.7	0.53	0.31	0.5
SumBasic	0.55	0.42	0.56	0.49	0.27	0.45
Proposed	0.74	0.64	0.7	0.62	0.37	0.54

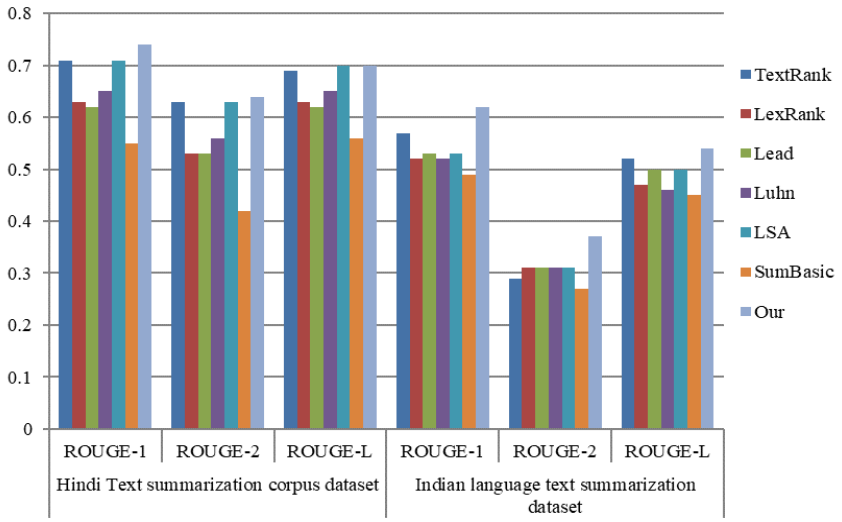

Fig. 10 ROUGE based comparison of proposed and other baseline approaches on Hindi text summarization corpus and ILSUM dataset for extractive text summarization

Table 7. ROUGE score for different datasets for Abstractive text summarization

Datasets	**English**			**Hindi**		
	ROUGE-1	**ROUGE-2**	**ROUGE-L**	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**
BBC News	0.80677	0.57429	0.73269	0.45556	0.27308	0.38148
CNN News	0.70576	0.55746	0.67545	0.35455	0.25625	0.32424
Hindi Text summarization corpus	_	_	_	0.33333	0.2963	0.26667
Indian language text summarization	0.68454	0.49604	0.61788	0.53692	0.22528	0.50121

3.5 Summary Evaluation

The proposed method for developing summaries in Hindi and English has been evaluated using ROUGE (Recall Oriented Understudy for Gisting Evaluation) [40].

We will also utilize several evaluation metrics, including as precision, recall, and F-measure, wich

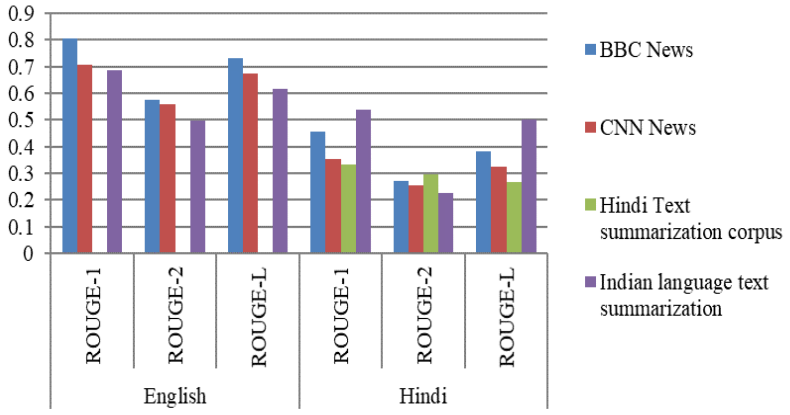

Fig. 11. ROUGE score for different datasets for Abstractive text summarization

Table 8. ROUGE score for multiple existing approaches of abstractive text summarization for BBC and CNN datasets in Hindi

Methods	BBC News dataset			CNN News dataset		
	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L
Sequence-to-Sequence	0.41	0.33	0.32	0.27	0.08	0.09
BERT	0.25	0.22	0.20	0.21	0.04	0.17
BART	0.14	0.10	0.17	0.11	0.04	0.08
PEGASUS	0.28	0.15	0.18	0.26	0.04	0.23
T5	0.34	0.28	0.34	0.41	0.19	0.30
Proposed	**0.66**	**0.61**	**0.37**	**0.67**	**0.56**	**0.59**

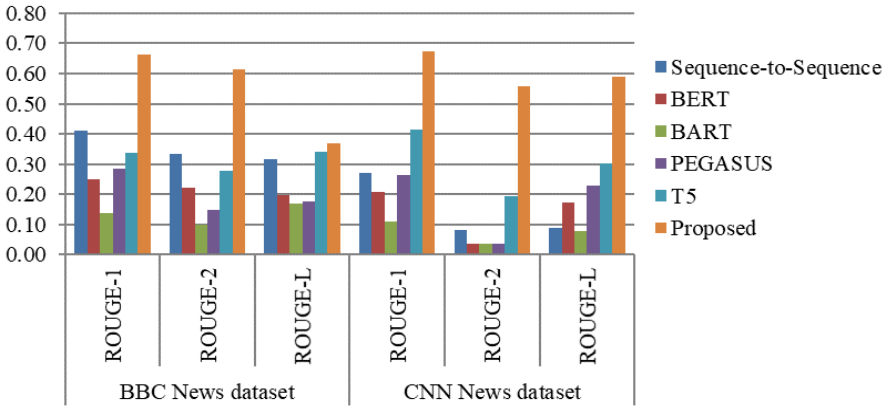

Fig. 12. ROUGE score for multiple existing approaches of abstractive text summarization for BBC and CNN datasets in Hindi

have previously been used to evaluated summaries in Hindi English.

These evaluation metrics are considered standard performance indicators for a system.

Table 9. ROUGE score for multiple existing approaches of abstractive text summarization for Hindi Text Summarization Corpus and ILSUM datasets

Methods	Hindi Text summarization corpus dataset			ILSUM dataset		
	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L
Sequence-to-Sequence	0.39	0.27	0.19	**0.27**	0.05	0.05
BERT	0.27	0.16	0.24	0.11	0.01	0.11
BART	0.18	0.05	0.09	0.09	0.03	0.07
PEGASUS	0.50	0.32	0.46	0.18	0.00	0.49
T5	0.31	0.32	0.38	0.23	0.15	**0.27**
Proposed	**0.59**	**0.34**	**0.56**	**0.27**	**0.22**	0.25

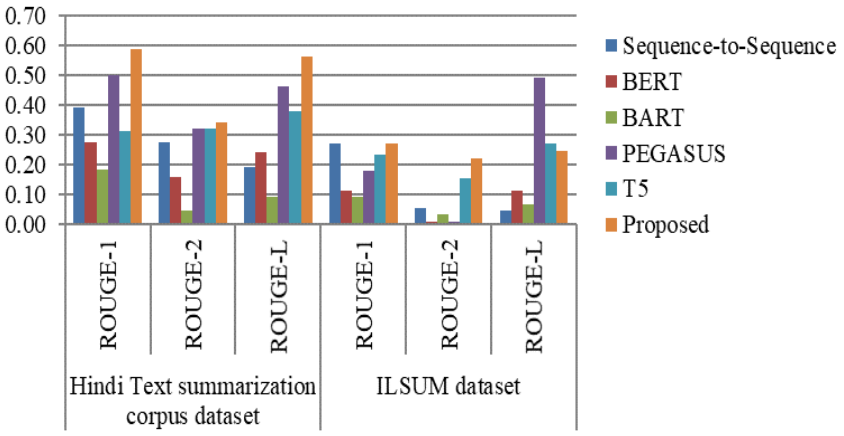

Fig. 13- ROUGE score for multiple existing approaches of abstractive text summarization for Hindi Text Summarization Corpus and ILSUM datasets

Datasets	Generated Extractive summary	Generated Abstractive summary
BBC News	Increase in the stamp duty threshold from £60,000 A freeze on petrol duty An extension of tax credit scheme for poorer families Possible help for pensioners The stamp duty threshold rise is intended to help first time buyers a likely theme of all three of the main parties' general election manifestos....	to the stamp duty gadget is a step in austria and was to be considered
CNN/Daily Mail	elections fill vacant seats rajya sabha held today members elected unopposed one seats facing by elections kerala mp resigned members uttar pradesh maharashtra bihar six members gujarat four.....	start explained details rajya sabha election held toda y end.
Hindi Text summarization corpus	पूर्व प्रधानमंत्री और कांग्रेस के दिग्गज नेता राजीव गांधी की आज 75वीं जयंती है. इस मौके पर कांग्रेस अध्यक्ष राहुल गांधी, यूपीए चेयरपर्सन सोनिया गांधी, प्रियंका गांधी.....	राजीव गांधी की पुण्यतिथि पर प्रधानमंत्री नरेंद्र मोदी ने दी श्रद्धांजलि
Indian language Text summarization corpus	यूपी में ब्राह्मण वोटरों को साधने की तैयारी कर रही है बीजेपी, लोगों से घर-घर जाकर मुलाकात करेंगे कें....	बीजेपी के केंद्र सरकार में ब्राह्मण मंत्रियों, सांसद और विधायक प्रबुद्ध ब्राह्मण समाज के लोगों से घर

Fig. 14. Generated multiple extractive and abstractive summaries

Methods	English summary	Hindi summary
Proposed	Commodore computer brand could be resurrected after being bought by a digital music distributor. new owner Yeahronimo Media Ventures has not ruled out the possibility of new breed of computers with the brand.	डिजिटल संगीत वितरक द्वारा खरीदे जाने के बाद कमोडोर कंप्यूटर ब्रांड को पुनर्जीवित किया जा सकता है। नए मालिक येहोनिमो मीडिया वेंचर्स ने ब्रांड के साथ कंप्यूटर की नई नस्ल की संभावना से इंकार नहीं किया है।
T5	The Commodore computer brand might come back to life after being acquired by a distributor of digital music. The prospect of a new breed of computers bearing the brand has not been discounted by the new owner, Yeahronimo Media Ventures.	डिजिटल संगीत के एक वितरक द्वारा अधिग्रहण के बाद कमोडोर कंप्यूटर ब्रांड फिर से जीवंत हो सकता है। ब्रांड वाले कंप्यूटरों की एक नई नस्ल की संभावना को नए मालिक, येहोनिमो मीडिया वेंचर्स द्वारा कम नहीं किया गया है।
PEGASUS	The once-famous Commodore computer brand could be resurrected after being bought by a US-based digital music distributor. Commore International filed for bankruptcy in 1994 and was sold to Dutch firm Tulip Computers. In the chronology of home computing,	एक समय के प्रसिद्ध कमोडोर कंप्यूटर ब्रांड को अमेरिका स्थित डिजिटल संगीत वितरक द्वारा खरीदे जाने के बाद पुनर्जीवित किया जा सका। कॉमोर इंटरनेशनल ने 1994 में दिवालियापन के लिए दायर किया और इसे डच फर्म ट्यूलिप कंप्यूटर्स को बेच दिया गया। होम कंप्यूटिंग के कालक्रम में
Seq-2-Seq	The brand of Commodore computers may resurrect after being purchased by a distributor of digital music. The new owner, Yeahronimo Media Ventures, hasn't ruled out the possibility of a new breed of computers bearing the name. The home computing pioneer Commodore may come back to life after being acquired by Yeahronimo Media Ventures, a distributor of digital music.	कमोडोर कंप्यूटर ब्रांड डिजिटल संगीत के एक वितरक द्वारा अधिग्रहित किए जाने के बाद जीवन में वापस आ सकता है। ब्रांड को प्रभावित करने वाले कंप्यूटर की एक नई नस्ल की संभावना को नए मालिक, येहरोनिमो मीडिया वेंचर्स द्वारा छूट नहीं दी गई है।
BERT	The once-famous Commodore computer brand may be brought back to life after being acquired by a digital music distributor with headquarters in the US. After declaring bankruptcy in 1994, Commore International was acquired by the Dutch company Tulip Computers. In the history of personal computers,	कमोडोर कंप्यूटर ब्रांड डिजिटल संगीत के एक वितरक द्वारा अधिग्रहित किए जाने के बाद जीवन में वापस आ सकता है। कंप्यूटर ब्रांड असर की एक नई नस्ल की संभावना नए मालिक, Yeahronimo मीडिया वेंचर्स द्वारा छूट नहीं दी गई है। कमोडोर, घर कंप्यूटिंग में अग्रणी, डिजिटल संगीत वितरक Yeahronimo मीडिया वेंचर्स द्वारा खरीदा जा रहा है के बाद पुनर्जीवित किया जा सकता है।
BART	Commodore, a pioneer in home computing, could be resurrected after being bought by digital music distributor Yeahronimo Media Ventures. The new owner plans to develop a "worldwide entertainment concept" with the brand, aiming to revive the Commodore 64 computer,	होम कंप्यूटिंग में अग्रणी कमोडोर को डिजिटल संगीत वितरक येहोनिमो मीडिया वेंचर्स द्वारा खरीदे जाने के बाद पुनर्जीवित किया जा सकता है। नए मालिक ने ब्रांड के साथ "विश्वव्यापी मनोरंजन अवधारणा" विकसित करने की योजना बनाई है, जिसका लक्ष्य कमोडोर 64 कंप्यूटर को पुनर्जीवित करना है,

Fig. 15. Generated abstractive summary for a document with proposed and other pre-trained transformer models

These metrics were computed using the equations (2), (3), and (4):

$$Precision\ (P) = \frac{No.of\ n-grams\ found\ in\ model\ \&\ reference}{No.of\ n-grams\ in\ model}, \quad (2)$$

$$Recall\ (R) = \frac{No.of\ n-grams\ found\ in\ model\ \&\ reference}{No.of\ n-grams\ in\ reference}, \quad (3)$$

$$F1-score = \frac{2\times P\times R}{P+R}. \quad (4)$$

To determine the ROUGE score, an N-gram score was produced, which is based on word and sequence overlap between the suggested summary and the reference summary, where N is the length of the document's N-grams (1, 2, 3, etc.).

We calculated three ROUGE scores for English and Hindi summaries: ROUGE-1, ROUGE-2, and ROUGE-L. The ROUGE-1, ROUGE-2, and ROUGE-L metrics are used to determine the similarity of unigrams, bigrams, and the longest common subsequence (LCS), respectively.

For comparison, we calculated the cosine similarity between the generated summary and the reference summary for extractive text summarizing. Equation (5) defines cosine similarity, which is utilized to compute content-based similarity metrics for generated summaries:

$$\text{Cosine similarity}\ (S_1, S_2) = \frac{S_1 .\ S_2}{||S_1||.||S_2||}, \quad (5)$$

where S1, S2 stands for the sentence's vectors. It is predicated on how the sentences overlap one another using a vector space model. Sentence rating is done after the similarity score is calculated, and the output summary consists of the sentences which are ranked highest [41].

4 Results and Discussions

The proposed model includes hyperparameters that are customized to the specific data set. The presented model incorporates hyper-parameters into both the label generation and summary generation stages. When the stored model is at its lowest loss factor, a call-back function is used to checkpoint the model after each epoch to see if the loss function has improved since the last best-saved instance.

For extractive text summarization in Hindi, Tables 5 and 6 define the different ROUGE scores of the proposed and other baseline approaches on BBC News, CNN News, the Hindi text summarization corpus, and ILSUM. It is obvious that the proposed method performs better overall than various methods like TextRank, LexRank, Lead, Luhn, LSA, and SumBasic. Figures 9 and 10 show the comparative analysis of different ROUGE scores between all the datasets. For extractive text summarization, TextRank and LSA yield the second-best results.

For abstractive text summarization, Table 7 defines the different ROUGE scores of the proposed approach in English and Hindi with respect to BBC News, CNN News, the Hindi text summarization corpus, and the ILSUM dataset.

The absence of the Hindi text summarization dataset in English is indicated by the blank score. Figure 11 shows the comparative analysis of different ROUGE scores between all the datasets. The results indicate that the English BBC News dataset has the highest ROUGE score of all the datasets; the Hindi ILSUM dataset has the highest ROUGE-1 and ROUGE-L scores; and the BBC dataset has the highest ROUGE-2 score.

The complete ROUGE scores for the CNN, BBC, and Hindi text summarization and ILSUM datasets are displayed in Tables 8 and 9, respectively. Similarly, the suggested method outperforms all other ROUGE scores across all datasets. The comparable results for the ILSUM dataset with ROUGE-1 and ROUGE-L are provided by Sequence to Sequence and T5. The comparative evaluation of the various ROUGE scores for Hindi across all datasets is displayed in Figures 12 and 13.

Figure 14 shows an example of a summary produced by the proposed approach for each of the input datasets. The comparison of the generated summaries with other current methods is shown in Figure 15. A document from the BBC dataset was used as a comparison, and summaries for each method were produced. This result shows the generated summaries, and the reference summary is most likely comparable.

5 Conclusions

In this paper, we present an approach for extractive and abstractive text summarization based on deep learning techniques, specifically the Seq2Seq model with RNN. We used this model to summarize the text document in both English and Hindi. The proposed method relied on translated text sources because Hindi datasets were unavailable.

We retrieved word embedding linguistic feature scores from each document and got the sentences for summary generation. The proposed study has been evaluated using different ROUGE metrics, such as ROUGE-1, ROUGE-2, and ROUGE-L. Also, we have computed other parameters, including precision, recall, and f-measure, across several datasets.

We used four different datasets, including BBC News articles and CNN News, to give summaries in English as well as in Hindi. We also employed two datasets containing exclusively Hindi documents: the Hindi Text Short Summarization Corpus and the ILSUM dataset.

We compared our methods to state-of-the art text summarization techniques such as TextRank, LexRank, Lead, Luhn, and SumBasic algorithms for extractive text summarization.

For abstractive text summarization, we have compared the proposed results with existing deep learning techniques such as sequence-to-sequence, BERT, BART, PEGASUS, and T5 transformers. The results clearly demonstrate the effectiveness of the proposed methodology in Hindi as well as in English text summaries.

References

1. **Agarwal, A., Naik, S., Sonawane, S. S. (2022).** Abstractive text summarization for

Hindi language using IndicBART. FIRE Working Notes, pp. 409–417.

2. **Mohammad-Masum, A. K., Abujar, S., Islam-Talukder, M. A., Azad-Rabby, A. S., Akhter-Hossain, S. (2019).** Abstractive method of text summarization with sequence to sequence RNNs. 2019 10th international conference on computing, communication and networking technologies, pp. 1–5. DOI: 10.1109/ICCCNT45670.2019.8944620.
3. **Alomari, A., Idris, N., Sabri, A. Q. M., Alsmadi, I. (2022).** Deep reinforcement and transfer learning for abstractive text summarization: A review. Computer Speech & Language, Vol. 71. DOI: 10.1016/j.csl.2021.101276.
4. **van-Lierde, H., Chow, T. W. (2019).** Learning with fuzzy hypergraphs: A topical approach to query-oriented text summarization. Information Sciences, Vol. 496, pp. 212–224. DOI: 10.1016/j.ins.2019.05.020.
5. **Naik, S. S., Gaonkar, M. N. (2017).** Extractive text summarization by feature-based sentence extraction using rule-based concept. 2017 2nd IEEE international conference on recent trends in electronics, Information & Communication Technology, pp. 1364-1368. DOI: 10.1109/RTEICT.2017.8256821.
6. **Khan, R., Qian, Y., Naeem, S. (2019).** Extractive based text summarization using k-means and TF-IDF. International Journal of Information Engineering and Electronic Business, Vol. 12, No. 3, p. 33. DOI: 10.5815/ijieeb.2019.03.05.
7. **Fang, C., Mu, D., Deng, Z., Wu, Z. (2017).** Word-sentence co-ranking for automatic extractive text summarization. Expert Systems with Applications, Vol. 72, pp. 189–195. DOI: 10.1016/j.eswa.2016.12.021.
8. **Ren, P., Chen, Z., Ren, Z., Wei, F., Nie, L., Ma, J., De-Rijke, M. (2018).** Sentence relations for extractive summarization with deep neural networks. ACM Transactions on Information Systems (TOIS), Vol. 36, No. 4, pp. 1–32. DOI: 10.1145/320086.
9. **Gulati, A. N., Sawarkar, S. D. (2017).** A novel technique for multidocument Hindi text summarization. 2017 international conference on nascent technologies in engineering. pp. 1–6. DOI: 10.1109/ICNTE.2017.7947890.
10. **Wu, Z., Lei, L., Li, G., Huang, H., Zheng, C., Chen, E., Xu, G. (2017).** A topic modeling based approach to novel document automatic summarization. Expert Systems with Applications, Vol. 84, pp. 12–23. DOI: 10.1016/j.eswa.2017.04.054.
11. **Alami, N., Mallahi, M. E., Amakdouf, H., Qjidaa, H. (2021).** Hybrid method for text summarization based on statistical and semantic treatment. Multimedia Tools and Applications, Vol. 80, pp. 19567–19600. DOI: 10.1007/s11042-021-10613-9.
12. **Azmi, A. M., Altmami, N. I. (2018).** An abstractive Arabic text summarizer with user controlled granularity. Information Processing & Management, Vol. 54, No. 6, pp. 903–921. DOI: 10.1016/j.ipm.2018.06.002.
13. **Sahoo, D., Bhoi, A., Balabantaray, R. C. (2018).** Hybrid approach to abstractive summarization. Procedia computer science, Vol. 132, pp. 1228–1237. DOI: 10.1016/j.procs.2018.05.038.
14. **Chopra, S., Auli, M., Rush, A. M. (2016).** Abstractive sentence summarization with attentive recurrent neural networks. Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies. pp. 93–98.
15. **Nallapati, R., Zhou, B., Gulcehre, C., Xiang, B. (2016).** Abstractive text summarization using sequence-to-sequence RNNS and beyond. DOI: 10.48550/arXiv.1602.06023.
16. **Zeng, W., Luo, W., Fidler, S., Urtasun, R. (2016).** Efficient summarization with read-again and copy mechanism. DOI: 10.48550/arXiv.1611.03382.
17. **See, A., Liu, P. J., Manning, C. D. (2017).** Get to the point: Summarization with pointer-generator networks. DOI: 10.48550/arXiv.1704.04368
18. **Cao, Z., Wei, F., Li, W., Li, S. (2018).** Faithful to the original: Fact aware neural abstractive summarization. Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32, No. 1. DOI: 10.1609/aaai.v32i1.11912.

19. **Zhang, Y., Li, D., Wang, Y., Fang, Y., Xiao, W. (2019).** Abstract text summarization with a convolutional seq2seq model. Applied Sciences, Vol. 9, No. 8, p. 1665. DOI: 10.3390/app9081665.
20. **Radford, A. (2018).** Improving language understanding by generative pre-training.
21. **Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L. (2018).** Deep contextualized word representations. Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Vol. 1, pp. 2227–2237. DOI: 10.18653/v1/N18-1202.
22. **Devlin, J. (2018).** Bert: Pre-training of deep bidirectional transformers for language understanding. DOI: 10.48550/arXiv.1810.04805.
23. **Roberts, A., Raffel, C., Lee, K., Matena, M., Shazeer, N., Liu, P. J., Zhou, Y. (2019).** Exploring the limits of transfer learning with a unified text-to-text transformer. Google Research.
24. **Dong, L., Yang, N., Wang, W., Wei, F., Liu, X., Wang, Y., Hon, H. W. (2019).** Unified language model pre-training for natural language understanding and generation. Advances in neural information processing systems, Vol. 32.
25. **Song, K., Tan, X., Qin, T., Lu, J., Liu, T. Y. (2019).** Mass: Masked sequence to sequence pre-training for language generation. arXiv: 1905.02450.
26. **Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L. (2019).** BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv1910.13461.
27. **Zhang, J., Zhao, Y., Saleh, M., Liu, P. (2020).** Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. International conference on machine learning, pp. 11328–11339.
28. **Qi, W., Yan, Y., Gong, Y., Liu, D., Duan, N., Chen, J., Zhang, R., Zhou, M. (2020).** Prophetnet: Predicting future N-gram for sequence-to-sequence pre-training. arXiv preprint arXiv:2001.04063.
29. **Bao, H., Dong, L., Wei, F., Wang, W., Yang, N., Liu, X., Hon, H. W. (2020).** Unilmv2: Pseudo-masked language models for unified language model pre-training. International conference on machine learning. pp. 642–652.
30. **Fedus, W., Zoph, B., Shazeer, N. (2022).** Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. Journal of Machine Learning Research, Vol. 23, No. 120, pp. 1–39.
31. **Zaheer, M., Guruganesh, G., Dubey, K. A., Ainslie, J., Alberti, C., Ontanon, S., Ahmed, A. (2020).** Big bird: Transformers for longer sequences. Advances in neural information processing systems, Vol. 33, pp. 17283–17297.
32. **Phang, J., Zhao, Y., Liu, P. J. (2022).** Investigating efficiently extending transformers for long input summarization. arXiv preprint arXiv:2208.04347.
33. **Gupta, P., Nigam, S., Singh, R. (2023).** A statistical language modeling framework for extractive summarization of text documents. SN Computer Science, Vol. 4, No. 6, p. 750. DOI: 10.1007/s42979-023-02241-x.
34. **Gupta, P., Nigam, S., Singh, R. (2023).** A statistical approach for extractive Hindi text summarization using machine translation. Proceedings of Fourth International Conference on Computer and Communication Technologies: IC3T 2022, Singapore: Springer Nature Singapore. pp. 275–282. DOI: 10.1007/978-981-19-8563-8_26.
35. **Shrivastava, M., Bhattacharyya, P. (2008).** Hindi POS tagger using naive stemming: harnessing morphological information without extensive linguistic knowledge. International Conference on NLP (ICON08), Pune, India.
36. **Porter, M. F. (1980).** An algorithm for suffix stripping. Program: electronic library and information systems. Vol. 14, No. 3, pp. 130–137. DOI: 10.1108/eb046814.
37. **Masum, A. K. M., Abujar, S., Talukder, M. A. I., Rabby, A. S. A., Hossain, S. A. (2019).** Abstractive method of text summarization with sequence to sequence RNNs. 2019 10th

international conference on computing, communication and networking technologies pp. 1–5. DOI: 10.1109/ICCCNT45670. 2019.8944620.

38. **Song, S., Huang, H., Ruan, T. (2019).** Abstractive text summarization using LSTM-CNN based deep learning. Multimedia Tools and Applications, Vol. 78, No. 1, pp. 857–875. DOI: 10.1007/s11042-018-5749-3.
39. **Chin-Yew, L. (2004).** Rouge: A package for automatic evaluation of summaries. Proceedings of the Workshop on Text Summarization Branches Out. pp. 74–81.
40. **Li, B., Han, L. (2013).** Distance weighted cosine similarity measure for text classification. Intelligent Data Engineering and Automated Learning – IDEAL 2013, IDEAL 2013., Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, Vol. 8206. DOI: 10.1007/978-3-642-41278-3_74.
41. **Mihalcea, R., Tarau, P. (2004).** Textrank: Bringing order into text. Proceedings of the 2004 conference on empirical methods in natural language processing, pp. 404–411.
42. **Erkan, G., Radev, D. R. (2004).** Lexrank: Graph-based lexical centrality as salience in text summarization. Journal of artificial intelligence research, Vol. 22, pp. 457-479. DOI: 10.1613/jair.1523.
43. **Mamidala, K. K., Sanampudi, S. K. (2021).** Text summarization for Indian languages: a survey. International Journal of Advanced Research in Engineering and Technology, Vol. 12, No. 1, pp. 530–538. DOI: 10.34218/ IJARET.12.1.2021.049.
44. **Wang, J., Liu, J., Wang, C. (2007).** Keyword Extraction Based on PageRank. In: Zhou, ZH., Li, H., Yang, Q. (eds) Advances in Knowledge Discovery and Data Mining, PAKDD 2007, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, Vol 4426. DOI: 10.1007/978-3-540-71701-0_95.

Article received on 30/04/2024; accepted on 07/08/2024.
**Corresponding author is Rajiv Singh.*