Journal of Institute of Science and Technology, 27(1), 141-148 (2022)
ISSN: 2467-9062 (print), e-ISSN: 2467-9240
https://doi.org/10.3126/jist.v27i1.46709

Research Article

ATTENTION BASED RECURRENT NEURAL NETWORK FOR NEPALI TEXT SUMMARIZATION

Bipin Timalsina, Nawaraj Paudel*, Tej Bahadur Shahi
Central Department of Computer Science and Information Technology, Tribhuvan University, Kathmandu, Nepal
**Corresponding Author: nawarajpaudel@cdcsit.edu.np*
(Received: *December 09, 2021;* ***Revised:*** *May 31, 2022;* ***Accepted:*** *June 30, 2022)*

ABSTRACT
Automatic text summarization has been a challenging topic in natural language processing (NLP) as it demands preserving important information while summarizing the large text into a summary. Extractive and abstractive text summarization are widely investigated approaches for text summarization. In extractive summarization, the important sentence from the large text is extracted and combined to create a summary whereas abstractive summarization creates a summary that is more focused on meaning, rather than content. Therefore, abstractive summarization gained more attention from researchers in the recent past. However, text summarization is still an untouched topic in the Nepali language. To this end, we proposed an abstractive text summarization for Nepali text. Here, we, first, create a Nepali text dataset by scraping Nepali news from the online news portals. Second, we design a deep learning-based text summarization model based on an encoder-decoder recurrent neural network with attention. More precisely, Long Short-Term Memory (LSTM) cells are used in the encoder and decoder layer. Third, we build nine different models by selecting various hyper-parameters such as the number of hidden layers and the number of nodes. Finally, we report the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) score for each model to evaluate their performance. Among nine different models created by adjusting different numbers of layers and hidden states, the model with a single-layer encoder and 256 hidden states outperformed all other models with F-Score values of 15.74, 3.29, and 15.21 for ROUGE-1 ROUGE-2 and ROUGE-L, respectively.

Keywords: Abstractive text summarization, recurrent neural network, long short term memory, encoder-decoder, Nepali language processing

INTRODUCTION
Text summarization is one of the challenging tasks in natural language processing, which aims to reduce the large text to its summary with essential content and meaning. It requires linguistic proficiency, world knowledge, and intelligence even performed manually. More technically, a summary is a reduced text which is produced from one or more large texts, that preserve important information in the original text(s), and is no longer than half of the original text(s) (Radev et al., 2014). Since the amount of textual information is prevalent on online media such as emails, news, blogs, and social media posts, the users are more interested to have summarized information for quick digestion, which can help them to make fast decisions. For instance, the summary of reviews about an online product can help customers to make a buying decision quicker. Such summarized information is not only beneficial for online business (Gaikwad & Mahender, 2016) but also useful for government service delivery, medical health informatics, and news agencies (Turpin et al., 2007; Adhikary et al., 2017) to generate a condensed summary. For example, search engines generate snippets as previews of the documents (Radev & Fan, 2000), and news websites produce condensed descriptions of news topics usually as headlines to facilitate news browsing (Adhikary et al., 2017), and so on.

Generally, text summarization can be achieved with two methods: extractive summarization and abstractive summarization. Here, extractive summarization is limited and extracts the important sentence or phrase from the source text without changing or modifying them to create a summary. Moreover, it doesn't change the order of sentences from the original text to the summary (Saranyamol & Sindhu, 2014). But, abstractive summarization is based on interpreting the original text, finding main concepts and relevant information, and expressing this information in the form of a summary (Gupta & Gupta, 2019). The summary generated with the abstractive summarization method will not merely select a few existing sentences from the original text but a compressed paraphrasing of the original content using vocabulary that might be unseen in the source documents (Nallapati et al., 2016; See et al., 2017). Therefore, the abstractive summary is more representative of the original text in comparison to the extractive summary. At the same time, it is more challenging to develop an automated abstractive summarization method as it requires good knowledge of the domain and natural language to understand the original text and represent it (Maharjan, 2020).

Although our focus on this work is text summarization for the Nepali language, we limit our discussion to the recent works carried out in other languages such as English considering that there are no existing works available for Nepali text summarization. Researchers proposed various machine learning and deep learning models for automated abstractive summarization for high-resource languages such as English (Yeasmin, et al., 2017; Banko, Mittal, & Witbrock, 2000). For instance, Song et al. (2019) proposed a neural network-based framework based on the combination of long short-term memory (LSTM) and convolutional neural network (CNN). In their method, the first stage extracts the phrases from the original text, and the second phase generates the summary. Their experiment on the Dailymail dataset produced a ROUGE score of 34.9% which outperforms other existing methods. A bidirectional gated recurrent unit (BiGRU) network is proposed for abstractive text summarization in the Indonesian language (Adelia et al., 2019). They considered two BiGRU models and reported the highest ROUGE score of 0.11 with a BiGRU model having 128 hidden units.

There are two main limitations in the aforementioned works. First, most of these works are investigated for high resource languages such as English which might not be appropriate for low resource languages such as Nepali. Second, they mostly used recurrent neural networks such as LSTM and GRU which might not capture the important concept during the training where the attention mechanism might help.

Given the limitation of existing works, we propose to build a Nepali text summarizer based on a recurrent neural network (RNN) with attention. For this, first, we built a text corpus that has both the original text and its summarized text. Here, we collect the news articles and their corresponding headlines from the online portals by web-scrapping. Second, we build nine different text summarization models based on a recurrent neural network with attention. Third, we evaluate the performance of these methods using widely used performance metrics, ROUGE, and report the best-performing model for Nepali text summarization.

In summary, our paper has the following main contributions.

a) We collect the news headlines and corresponding news content from the web and prepare a text summarization corpus.
b) To the best of our knowledge, this is the first work on an abstractive text summarization of Nepali text documents.
c) We propose nine different text summarization models based on a recurrent neural network with attention.
d) We evaluate and compare the proposed models using widely used performance metrics, ROUGE, and report the best-performing model for benchmarking.

The rest of the paper is organized as follows. Section "Related work" discusses the recent work carried out on text summarization. The workflow of the proposed method is discussed in the section "Methodology". The Section "Experiment and Results" reports the detail about experiments and results on Nepali text summarization. Finally, the "Conclusion and Future Work" section concludes our paper.

RELATED WORK

With the rise of deep learning methods in various applications of natural language processing such as text classification (Subba et al., 2019), and sentiment analysis (Sitaula et al., 2021), researchers have been investigating the effectiveness of deep learning methods on abstractive approach to text summarization (Liu et al., 2018; Adelia et al., 2019). The representation of data with multiple levels of abstraction can be learned with deep learning models that are composed of multiple processing layers and computational nodes (LeCun et al., 2015).

The research on automatic text summarization can be traced back to six decades ago. Most early works on summarization focused on technical documents like generating abstracts from the research papers. Baxendale et.al. (1958) proposed a positional method where the first and last sentences of paragraphs are considered as summary sentences (Baxendale, 1958). The frequency of words and phrases in a document is used in the automatic text summarization technique developed by Luhn (1958). The author also performed data preprocessing like stemming and stop word removal before summarization. The sentences with the highest concentrations of salient content terms are considered as a summary in Luhn's method. Another method for summarization has focused on the presence of high-frequency content words (keywords), pragmatic words (cue words), title and heading words, and structural indicators (sentence location) for extractive summarization tasks Edmundson, 1969). A knowledge-based summarization system called FRUMP (Fast Reading Understanding and Memory Program) has used a template-filling approach to news stories (DeJong, 1977). Naïve Bayes Classifier and sentence scoring features can be used to generate a summary using a trainable document summarizer (Kupiec et al., 1995). It has been found that the maximum entropy classifier outperforms the Naïve Bayes approach (Osborne, 2002). The diversity-based ranking for reordering documents and producing summaries can be done using maximal marginal relevance (Carbonell & Goldstein, 1998). This approach is for generating a query-based summary. A centroid-based text summarization technique works for single as well as multi-document summarization (Radev et al., 2004). A popular extractive text summarization technique called LexRank is an unsupervised approach to text summarization based on graph-based centrality scoring of sentences (Erkan & Radev, 2004).

Besides extractive text summarization, there is increasing attention from researchers on abstractive text summarization in the recent past (Yeasmin et al., 2017), especially after the success of deep neural networks in various applications such as computer vision and image processing (Mishra & Shahi, 2021). Initially, the text summarization task was investigated to generate news headline generation, abstract of research papers, and so on (Edmundson, 1969; DeJong, 1977; Carbonell & Goldstein, 1998). A text summarization method inspired by statistical machine translation was proposed by Banko et al. (2000) using a news corpus of headline-article pairs. They generated the headlines for the news article even shorter than one sentence using statistical term selection and term ordering jointly which their model learns directly from the training corpus. Similarly, a neural network-based approach with a larger dataset of headline-article pairs was implemented by Rush et al., 2015). They achieve state-of-the-art performance on both DUC-2004 and Gigaword datasets which have two sentence level summaries. A further performance boost was reported in Nallapati et al. (2016) using a sequence to a sequence-based attentional encoder-decoder neural network. This work was based on an attentional recurrent neural network implemented for a machine translation task (Bahdanau et al., 2014). Furthermore, an LSTM network with attention was implemented for news headline generation (Lopyrev, 2015)

A few works on text summarization using the generative adversarial network are also reported in the literature. For instance, a text summarization using GAN is proposed by Lin (2004). In this work, they have designed and trained both a generator and discriminator in an end-to-end fashion. The generator works as an agent of reinforcement learning, taking the raw text as input and predicting the abstractive summary whereas the discriminator attempts to distinguish the generated summary from the human-generated summary. Their experiments have concluded the model achieves competitive ROUGE scores with the state-of-the-art methods on CNN/Daily Mail dataset.

METHODOLOGY

The methodology includes implementing the Recurrent Neural Network with LSTM units and Attention. The model is trained using the data collected from online Nepali news portals. The headlines and corresponding news articles are used from the dataset for training and testing purposes. The ROUGE score of the model is computed based on the automatically generated headline (one-line summary) as output.

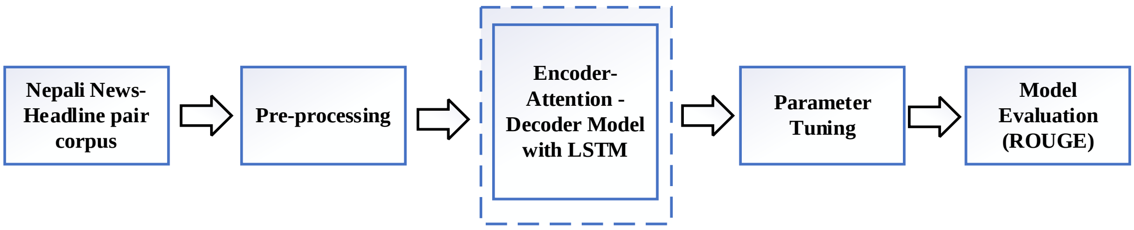

Fig.1: The overall workflow of the proposed method for Nepali text summarization

Data Collection and Cleaning

Since an appropriate dataset having Nepali text and summary is unavailable, a new dataset is prepared during this study. Nepali News dataset with news articles and headlines is generated by scrapping the Nepali online news portals. The collected data is preprocessed by removing different unwanted characters and Hypertext markup language (HTML) tags to get clean Nepali news headline-article pairs.

Encoder – Attention – Decoder Model with Long Short Term Memory

The bidirectional LSTM consists of the forward and backward LSTMs. The forward LSTM $\vec{f}$ reads the input text sequence as $\{x_1, x_2, \dots x_t\}$ and calculates a sequence of forward hidden states $\{\overrightarrow{h_1}, \overrightarrow{h_2}, \dots, \overrightarrow{h_t}\}$. The backward LSTM $\overleftarrow{f}$ reads the sequence in the reverse order $\{x_t, x_{t-1}, \dots x_1\}$ resulting in a sequence of backward hidden states $\{\overleftarrow{h_t}, \overleftarrow{h_{t-1}}, \dots, \overleftarrow{h_1}\}$. Finally, an annotation for word x_j is obtained by concatenating these two hidden states (Eq. 1) to generate summaries of both the precessing words and the following words.

$$h_j = \left[\overrightarrow{h_j}^T ; \overleftarrow{h_j}^T \right]^T \qquad Eq.(1)$$

Now, all the vectors h_1, h_2, h_3, …., h_t are representations of 't' number of tokens in the input sentence. Here, we took a weighted sum of all hidden states to represent a context vector 'c' instead of considering the last hidden state (h_t) as a context vector as suggested by Bahadanau et al. (2014). This approach emphasises embedding all the words in the input while creating the context vector which is essential to generate a more representative summary.

Finally, the decoder is trained to predict the next word y_t' given the context vector '*c*' and all the previously predicted words. In this model, each conditional probability is defined as in Eq. (2)

$$p(y_i | y_{1,}, \cdots, y_{i-1}, x) = g(y_i - 1, s_i, c_i) \quad Eq.(2)$$

where g is a nonlinear activation function that outputs the probability of y_i and s_i is an LSTM hidden state for time i, computed with Eq. (3)

$$s_i = f(s_{i-1}, y_{i-1}, c_i) \qquad Eq.\ (3)$$

Here the probability is conditioned on a distinct context vector c_i for each target word y_i.

The context vector c_i is, calculated as the weighted sum of these annotations h_i as defined in Eq. 4:

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \qquad Eq.(4)$$

The weight α_{ij} of each annotation, h_j is computed with Eq. (5)

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{T_x} exp(e_{ik})} \qquad Eq.(5)$$

where

$$e_{ij} = \mathrm{a}(s_{i-1}, h_j)$$

Here e_{ij} is the output score of a neural network described by the function a which seeks to capture the alignment between input at j and output at i.

Evaluation metrics

We used **ROUGE**, a widely used performance metric to measure the correctness of the proposed summarizer. The **ROUGE** is a set of metrics for evaluating automatic summarization of texts as well as machine translation. It works by comparing an automatically produced summary or translation against a set of reference summaries. The quantitative measure of **ROUGE** counts the number of overlapping units such as n-gram, word sequences, and word pairs between the systems-generated summary and the gold standard summaries by a human expert. This evaluation metric was introduced by Chin-Yew Lin describing four different measures of **ROUGE** measures: **ROUGE**-N, **ROUGE**-L, **ROUGE**-W, and **ROUGE**-S (Lin, 2004).

Recall in the context of **ROUGE** gives how much of the reference summary the system summary recovering is. Considering the individual words, it can be calculated as:

$$\text{Recall (R)} = \frac{\text{number\_of\_overlapping\_words}}{\text{total\_words\_in\_reference\_summary}} \qquad Eq.(6)$$

Similarly, Precision in **ROUGE**, means how much of the system summary is relevant. In the context of ROUGE precision is computed as:

$$Precision\ (P) = \frac{\text{number\_of\_overlapping\_words}}{\text{total\_words\_in\_system\_summary}} \qquad Eq.(7)$$

F-Score is the harmonic mean of Precision and Recall. The value of F-Score can be obtained by using the following expression:

$$\text{F} - \text{Score (F)} = 2 \times \frac{\text{Precision } \times \text{Recall}}{\text{Precision} + \text{Recall}} \qquad Eq.(8)$$

Among these four ROUGE measures, we used two measures: **ROUGE-N** and **ROUGE**-L. Here **ROUGE**-N measures unigram, bigram, trigram, and higher-order n-gram overlap, whereas **ROUGE-L** measures the longest matching sequence of words. More specifically, we report the results for **ROUGE**-1, **ROUGE**-2, and **ROUGE**-L to evaluate the performance of the proposed Nepali text summarizer.

Implementation

The text preprocessing, model building, and evaluation metrics calculation were implemented with Python programming language. The Keras with TensorFlow is used to implement the recurrent neural network models, whereas other libraries like Pandas, NumPy, and Matplotlib were used for data processing and visualization. We trained our models on Google Colab with NVIDIA K80 graphical processing unit (GPU) of 12 GB RAM provided by Google for free.

EXPERIMENTS AND RESULTS

Data

The data was preprocessed by removing numeric characters and brackets. The data set contains around 117 thousand (1, 17,566) Nepali headline-news pairs. Two percentage of data was used for testing. The remaining data were used during training the models where 10 % of them were used for validation. The first 180 words (less if the article does not have many) of each article and 10 words (less if the headline does not have many) of each headline were selected for the experiment. The numbers of unique tokens for training articles and headlines are found to be 4, 75,964, and 66,554 respectively. Separate vocabularies for articles and headlines were created for the experiments.

Different Models and Parameters

In this research work, Attention-based Recurrent Neural Networks with Long Short Term Memory (LSTM) were developed for Nepali Text Summarization. The purpose of the research was to develop an abstractive Nepali Text Summarizer by implementing encoder-decoder RNN. In this study, LSTM-based sequence-to-sequence models with attention were developed, trained, and tested for ROUGE scores. The decoder was fixed to be a single-layered LSTM in all the models whereas single-layered, 2-layered and 3-layered LSTMs were used as an encoder. Since there is no existing work for a Nepali text summarizer to compare, we assume the single-layered model with 128 hidden units (Model_A1) as the baseline model (Ref to Table 2).

The maximum sequence length for input (news article) and target (corresponding headline) was set to 180 and 10 words respectively. The target sequences were padded with 'sostk' as the start of the sequence token and 'eostk' as the end of the sequence token. After converting texts into numeric sequences, padding was performed for the sequences with sequence lengths less than that of maximum sequence lengths.

We find the best set of hyperparameters for each model listed in Table 1 using a random search approach. The word embedding dimension was set to 200 and hidden units were in the range of 128, 256, and 512 in each model except in the "Model_C3" due to memory constraints. RMSprop was used as an optimizer with a learning rate of 0.001 and "Sparse Categorical Crossentropy" was set as a loss function while training the model. Model overfitting was controlled with dropout and early stopping. The training was run up to 60 epochs where early stopping was used along with a patience level of 7.

Table 1. Detailed hyper-parameters used in our study

Parameters	Value
Embeding Size	200
Batch Size	256
Dropout (encoder)	0.5
Dropout (decoder)	0.3
Learning rate	0.001
Optimizer	RMSProp
Loss function	Sparse Categorical Crossentropy
Epoch	60

RESULTS AND ANALYSIS

The overall ROUGE Scores in terms of precision, recall, and F-Scores for all the models are represented in Table 2 which summarizes the overall experiments and results.

Table 2. ROUGE Scores of nine different models

S.N.	Model Name	No. of Layers		No. of Hidden Units	ROUGE Scores								
		Encoder	Decoder		ROUGE-1			ROUGE-2			ROUGE-L		
					P	R	F	P	R	F	P	R	F
1	**Model_A1**	1	1	128	14.89	12.43	13.08	3.28	2.23	2.55	14.67	12.19	12.87
2	**Model_A2**	**1**	**1**	**256**	**18.00**	**14.99**	**15.74**	**4.14**	**3.07**	**3.29**	**17.35**	**14.53**	**15.21**
3	**Model_A3**	1	1	512	17.84	15.00	15.49	3.16	2.69	2.26	17.04	14.33	14.78
4	**Model_B1**	2	1	128	12.76	10.63	11.06	1.86	1.32	1.49	12.54	10.50	10.91
5	**Model_B2**	2	1	256	16.39	12.93	13.87	3.47	2.37	2.74	16.18	12.80	13.70
6	**Model_B3**	2	1	512	14.79	11.91	12.67	3.11	2.19	2.46	14.33	11.52	12.26
7	**Model_C1**	3	1	128	11.00	8.43	9.11	1.45	1.15	1.18	10.79	8.31	8.96
8	**Model_C2**	3	1	256	14.10	11.02	11.81	3.40	2.17	2.48	13.93	10.87	11.65
9	**Model_C3**	3	1	300	14.27	10.45	11.42	2.57	1.97	2.06	13.88	10.22	11.13

Among these nine models, a single-layered model with 256 hidden dimensions (Model_A2) has outperformed in terms of ROUGE -scores. Some of the summaries generated by this model are excellent both grammatically and semantically but some outputs are opposing and out of context also. Some sentences generated by the model are conveying the meanings but with improper structures having repeated words, missing words, and grammatical errors. This type of pattern of headline generation is similar in all the models but with different ROUGE Scores. Some samples of the machine-generated headlines (summaries) are expressed in Table 3 along with articles and human-generated headlines.

Table 3. Sample outputs generated by summarizer model (Model_A2) with best ROUGE Scores

S. N.	Article	Machine-generated Headline	Human-generated Headline
1.	भारी वर्षाका कारण पासाङ ल्हामु मार्गको त्रिशूली नजिक रहेको ढुंगेबजार–बेत्रावती सडक खण्डमा पर्ने तिरतिरे भीर र गुह्य खोल्सामा सडक अवरुद्ध भएको छ । सडक खुलाउन प्रयास भइरहेको प्रमुख जिल्ला अधिकारी द्रोण पोखरेलले बताए । ठाउँ–ठाउँमा पहिरो गएकाले सडक खुलाउन स्थानीय सशस्त्र र नेपाल प्रहरी परिचालित भएको उनले बताए । . . .	पहिरोले सडक अवरुद्ध	पहिरोले सडक अवरुद्ध
2.	कुनै बेला कैलालीको बडका वनभित्र हराएको व्यत्ति फर्केर आउला भन्ने आशा कमैले राख्थे । अहिले त्यो जंगल छर्लंग देखिने भएको छ । बूढो बांगोटिंगो र ढुंग्रो परेका पुनरुत्पादन सालको जंगल छैन । हजार बढी हेक्टर अतिक्रमणको चपेटामा परेको बडका वनको अहिलेको क्षेत्रफल हजार जति मात्रै छ भारतको दुदुवा नेसनल पार्क जोड्ने बडका वनको भजनी र क्षेत्र लगभग सकिएको अवस्थामा छ । असिना र जंगलभित्र उपयोगी साल मुस्किलले भेटिने गरेको इलाका वन कार्यालय भजनीका सहायक वन अधिकृत हरिलाल यादव बताउँछन् । . . .	चुरे निर्माणमा भोट	यसरी नासियो बडका वन
3.	साफ च्याम्पियनसिपमा सन् पछि नेपाल मा मात्र सेमिफाइनल पुग्यो । गोवामा नेपालले तय गरेको अन्तिम चारको यात्रा गर्व गर्न लायक भने थिएन ।यस्तोमा दक्षिण एसियाली फुटबलमै नेपालका लागि सेमिफाइनल एक प्रकारले बनेको थियो । प्रत्येक साफ च्याम्पियनसिप अगाडि प्रमुख प्रश्न यही बनेको थियो के नेपाल यसपल्ट सेमिफाइनल पुग्नेछ त सन् र को दुई संस्करणमा भने नेपाल लगातार सेमिफाइनलको यात्रा तय गर्न सफल रह्यो । . . .	साफ च्याम्पियनसिप नेपाल सेमिफाइनलमा	लगातारको सेमिफाइनल यात्रा
4.	दसैंका बेला अनुगमन र कर्मचारीतन्त्र सुस्ताएको मौका छोपेर त्रिशूली दोहन र नदीजन्य उत्खनन बढेको छ । जिल्ला अनुगमन समितिले रोक लगाएका दर्जनौं अवैध बालुवा खानीहरू रातारात सञ्चालनमा छन् । गिट्टी कुट्ने उद्योगहरूका लागि रातारात त्रिशूलीमा उपकरण लगाएर ढुङ्गा झिक्ने क्रम पनि रोकिएको छैन । जिल्ला अनुगमन समितिले गत हप्ताअघि मात्रै बालुवा उत्खनन र निकासीमा रोक लगाएका रातभर सञ्चालनमा छन् । . . .	बालुवा र बालुवा तस्करी	त्रिशूलीमा दोहन बढ्यो
5.	यसपालि समयमै मनसुन नआउँदा पर्सामा धान रोपाइँ ढिला भयो। एकसाताअघि मात्र रोपाइँ भ्याएका किसान अहिले बन्दको मारमा परेका छन्। सीमांकनको विरुद्धमा मधेसवादी दलले अनिश्चितकालीन बन्दका कारण रासायनिक मलको हाहाकार मच्चिएको छ। अभावसँगै मलको कालोबजारी समेत सुरु भएको किसानले गुनासो गरेका छन्। . . .	मल अभावले किसान प्रभावित	कृषकलाई मल अभाव

CONCLUSION AND RECOMMENDATIONS

Abstractive text summarization can be done by using deep learning methods. The encoder-decoder RNN model was initially developed for machine translation tasks. This approach can be used for solving text summarization problems. In this study, encoder-decoder RNN with LSTM units and attention is used to develop some experimental models. Different models are constructed using a single-layer decoder and encoder layers are varied to be single-layered, double-layered, and triple-layered. The experiments are conducted with 128,256 and 512 hidden units of LSTM cells. Data for the experiment are collected from online Nepali news portals. Among these models, the model with a single-layer encoder and 256 hidden units outperformed. The ROUGE-1 F-Score for the model is 15.74. Similarly, ROUGE-2 and ROUGE-L F-Scores of the model are 3.29 and 15.21 respectively. The quantitative, as well as qualitative performance of the model, is satisfactory. Since there are not any baseline works for abstractive Nepali text summarization, this work may act as a baseline for future research.

This work is simply a beginning toward abstractive Nepali text summarization. There are many other approaches and techniques which can be implemented to deal with the problem. Being specific to the approach applied in this work, some ideas may help to achieve more quantitative and qualitative results than this. One of them is increasing the data size, because, it is believed that deep learning models learn better and perform well if they are trained with many data. The system may result better if a bi-directional encoder is used since it is capable of capturing the context from both directions and results in a better context vector. The beam search strategy can be used for decoding the test sequence instead of using the greedy approach as in this study. This may help to produce more qualitative word sequences. Other approaches like pointer generation with coverage mechanism can be used to produce improved results

ACKNOWLEDGEMENTS

The authors would like to thank Prof. Dr. Shashidhar Ram Joshi, Prof. Dr. Subarna Shakya, and all other faculty members and staff of the Central Department of Computer Science and Information Technology, Tribhuvan University.

AUTHOR CONTRIBUTION STATEMENTS

BT: contributed for data curation, analysis, software, validation, visualization, writing – original draft, writing – review & editing; NP: contributed for supervision, writing – review & editing; TBS: contributed for analysis, validation, visualization, writing – original draft, writing – review & editing

CONFLICT OF INTEREST

The authors confirm no conflict of interests.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author, upon reasonable request.

REFERENCES

Adelia, R., Suyanto, S., & Wisesty, U.N. (2019). Indonesian abstractive text summarization using bidirectional gated recurrent unit. *Procedia Computer Science, 157*, 581-588.

Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E.D., Gutierrez, J.B., & Kochut, K. (2017). Text summarization techniques: A brief survey. *arXiv*. https://doi.org/10.48550/arXiv.1707.02268.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv*, https://doi.org/10.48550/arXiv.1409.0473.

Banko, M., Mittal, V.O., & Witbrock, M.J. (2000). Headline generation based on statistical translation. *Proceedings of the 38th annual meeting on association for computational linguistics* (pp. 318-325). Hong Kong.

Baxendale, P. (1958). Machine-made index for technical literature: An experiment. *IBM Journal of Research and Development, 2*(4), 354-361.

Brownlee, J. (2017). Machine learning mastery. Retrieved September 20, 2019, from https://machinelearningmastery.com/encoder-decoder-deep-learning-models-text-summarization/

Carbonell, J., & Goldstein, J. (1998). The use of MMR, diversity-based reranking for reordering documents and producing summaries. *21st annual international ACM SIGIR conference on research and development in information retrieval*. Melbourne, Australia.

DeJong, G. (1977). *Skimming newspaper stories by computer*. Ph.D. Thesis, Yale University .

Edmundson, H.P. (1969). New methods in automatic extracting. *Journal of the Association for Computing Machinery, 16*(2), 264-285.

Erkan, G., & Radev, D.R. (2004). LexRank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research, 22*, 457-479.

Gaikwad, D.K., & Mahender, C.N. (2016). A review paper on text summarization. *International Journal of Advanced Research in Computer and Communication Engineering, 5*(3), 154-160.

Gupta, S., & Gupta, S. (2019). Abstractive summarization: An overview of the state of the art. *Expert Systems with Applications, 121*, 49-65.

Kasture, N.R., Yargal, N., Singh, N.N., Kulkarni, N., & Mathur, V. (2014). A Survey on methods of abstractive text summarization. *International Journal for Research in Emerging Science and Technology, 1*(6), 53-57.

Kupiec, J., Pedersen, J., & Chen, F. (1995). A trainable document summarizer. *18th annual international ACM SIGIR conference on research and development in information retrieval*. Washington, USA.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436–444.

Lin, C.Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Text summarization branches out* (pp. 74-81). Association for Computational Linguistics, Barcelona, Spain.

Liu, L., Lu, Y., Yang, M., Qu, Q., Zhu, J., & Li, H. (2018). Generative adversarial network for abstractive text summarization. *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, 32*(1), 8109-8110.

Lopyrev, K. (2015). Generating news headlines with Recurrent Neural Networks. *arXiv*, https://doi.org/10.48550/arXiv.1512.01712.

Luhn, H.P. (1958). The automatic creation of literature abstracts. *IBM Journal of Research and Development, 2*(2), 159-165.

Maharjan, R. (2020). *Abstractive text summarization using recurrent neural network with attention based mechanism*. Masters' Thesis, Tribhuvan University, Central Department of Computer Science and IT, Kathmandu, Nepal.

Mishra, B., & Shahi, T.B. (2021). Deep learning-based framework for spatiotemporal data fusion: an instance of landsat 8 and sentinel 2 NDVI. *Journal of Applied Remote Sensing, 15*(3), 034520. https://doi.org/10.1117/1.JRS.15.034520.

Nallapati, R., Zhou, B., & Santos, C.D. (2016). Abstractive text summarization using sequence-to-sequence RNNs and beyond. *20th SIGNLL conference on computational natural language learning (CoNLL)*, Berlin, Germany.

Osborne, M. (2002). Using maximum entropy for sentence extraction. *Proceedings of the Association for Computational Linguistics (ACL)-02 workshop on automatic summarization*. Phildadelphia, Pennsylvania, USA.

Radev, D.R., Hovy, E., & McKeown, K. (2002). Introduction to the special issue on summerization. *Computational Linguistics, 28*(4), 399-408.

Radev, D., & Fan, W. (2000). Automatic summarization of search engine hit lists. *Workshop on recent advances in natural language processing and information retrieval* (pp. 99-109). Hong Kong.

Radev, D., Allison, T., Blair-Goldensohn, S., Blitzer, J., Celebi, A., Dimitrov, S., Drabek, E., Hakim, A., Lam, W., Liu, D., Otterbacher, J., Qi, H., Saggion, H., & Zhang, Z. (2004). MEAD-a platform for multidocument multilingual text summarization. *Proceedings of the 4th international conference on language resources and evaluation* (pp. 699-702), Lisbon, Portugal.

Rush, A.M., Chopra, S., & Weston, J. (2015). A neural attention model for abstractive sentence summarization. *Proceedings of the 2015 conference on empirical methods in natural language processing* (pp. 379-389), Lisbon, Portugal.

Saranyamol, C., & Sindhu, L. (2014). A survey on automatic text summarization. *International Journal of Computer Science and Information Technologies, 5*(6), 7889-7893.

See, A., Liu, P.J., & Manning, C.D. (2017). Get to the point: summarization with pointer-generator networks. *arVix*, https://doi.org/10.48550/arXiv.1

704.04368 .

Sitaula, C., Basnet, A., Mainali, A., & Shahi, T. (2021). Deep learning-based methods for sentiment analysis on Nepali COVID-19 related tweets. *Computational Intelligence and Neuroscience*, 2158184, https://doi.org/10.1155/2021/2158184.

Song, S., Huang, H., & Ruan, T. (2019). Abstractive text summarization using LSTM-CNN based deep learning. *Multimedia Tools and Applications, 78*, 857-875.

Subba, S., Shahi, T., & Paudel, N. (2019). Nepali text document classification using beep neural network. *Tribhuvan University Journal, 33*(1), 11-22.

Turpin, A., Tsegay, Y., Hawking, D., & Williams, H.E. (2007). Fast generation of result snippets in web search. *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 127-134), Amsterdam, The Netherlands.

Yeasmin, S., Tumpa, P.B., Nitu, A.M., Uddin, M.P., Ali, E., & Afjal, M.I. (2017). Study of abstractive text summarization techniques. *American Journal of Engineering Research, 6*(8), 253-260.