INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

Text Summarization in Assamese Language using Sequence to Sequence RNNs

Pritom Jyoti Goutom[1]*, **Nomi Baruah**[2]

1 Center for computer Science and Application, Dibrugarh University, Dibrugarh, Assam, India
2 Department of Computer Science and Engineering, DUIET, Dibrugarh University, Dibrugarh, Assam, India

OPEN ACCESS

Received: 23-03-2023

Accepted: 26-06-2023

Published: 02-11-2023

Editor: Guest Editors: Dr. Madhuryya Saikia & Dr. Niranjan Bora

Citation: Goutom PJ, Baruah N (2023) Text Summarization in Assamese Language using Sequence to Sequence RNNs. Indian Journal of Science and Technology 16(SP2): 22-29. https://doi.org/10.17485/IJST/v16iSP2.5429

* **Corresponding author**.

pritomjyotigoutom1996@gmail.com

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

This study discusses an approach for abstractive text summarization in the Assamese language. **Objective:** The main objective of this paper is to develop a novel approach for abstractive text summarization in Assamese that efficiently condenses large information while keeping its core meaning. **Methods:** We utilise a sequence-to-sequence Recurrent Neural Network [RNN] model with an encoder-decoder architecture in this paper. In this study, we use a Bi-LSTM on the encoder side, an attention mechanism, a softmax layer on the decoder side, and Assamese news items obtained from renowned daily newspapers in Assam. **Findings:** The results show that the suggested model is effective, with a train loss of 0.008 and assessment scores based on ROUGE-1, ROUGE-2, and ROUGE-L criteria. **Novelty:** The novelty lies in filling the aforementioned gap by proposing and implementing an abstractive text summarization approach for the Assamese language. Improve Assamese text summarization approaches by applying the presented strategy to news items.

Keywords: Abstractive; NLP; RNN; Seq2Seq; Text summarization

1 Introduction

Summarization is a critical component of natural language discourse, and it is one of the most challenging challenges in artificial language. A person can easily mechanically summarize any phrase or text [1]. They choose the key phrases from a book and create the summary. The task is far more challenging on a computer system. Due to the fact that summarizing transforms text, we have adapted that approach to summarize Assamese text.

A summary of a document can be written in a variety of ways, but there are two main types that can be distinguished based on the information chosen and the way it is organised: extractive and abstractive approaches. In essence, utilising features, extractive summarization identifies the key phrases in the text and groups them to create the summary [2]. It is comparable to underlining text with a highlighter. Instead, than choosing the key phrases from the original material that contain the most important information, abstractive summarization creates new sentences from scratch. Like a person writing with a pen a summary of what he has been thinking. Tools for summarizing information based on machine learning are currently accessible. But it's challenging to locate language-specific models [3].

The summary of a text, which is critical to efficient communication, is one of the key problems in natural language processing [4]. While people may readily construct summaries by extracting essential phrases, automating the same operation with computers is far more difficult. In this study, we adjusted existing approaches to meet the summary demands of Assamese literature, taking into account its distinctive linguistic peculiarities.

The existing approaches for text summarization can be broadly classified into extractive and abstractive methods based on how they select and organize information [5]. Extractive summarization involves identifying significant words or phrases from the source text and arranging them to form a summary, similar to using a highlighter to underline text. On the other hand, abstractive summarization constructs new sentences that capture the essential information, rather than relying solely on extracted keywords. Currently, machine learning-based tools are available for information summarization, but finding language-specific models remains a challenge.

In this paper, we present a comprehensive method for autonomously extracting news articles from two reputable Assamese publications and generating meaningful summaries. To ensure high-quality results, we employ various pre-processing techniques such as word embedding, vocabulary counting, and missing word analysis to cleanse and refine the input texts and summaries [6]. Additionally, we train a Word2vec model specifically designed to generate Assamese word embedding using our dataset.

Our proposed approach incorporates a two-layered bidirectional recurrent neural network (RNN) with LSTM cells and the Bahdanau attention model [7]. The encoder processes the input phrases and produces a fixed-length vector, which serves as the foundation for the decoder to generate the output sequence. We have also modified our model to handle the summarization of Assamese text that has been previously translated via machine translation.

To address the limitations of existing approaches, our study focuses on several crucial aspects of text summarization, aiming to enhance productivity and create more fluid and effective summaries. While we provide detailed descriptions of important processes, the primary emphasis of this paper is on explaining the deep learning models and methodologies that underlie our proposed approach.

In summary, this study presents a novel method for summarizing Assamese text, overcoming the limitations of existing approaches. By combining pre-processing techniques, language-specific word embedding, and a well-designed sequence-to-sequence model, we aim to produce accurate and coherent summaries [8–10]. The subsequent sections of this paper will provide a detailed explanation of our methodology and present the evaluation results.

Here are a few potential research gaps we could explore

1. **Model Architecture:** The present studies used LSTM for abstractive text summarization in Assamese. We mentioned utilising the Bi-LSTM method. We may evaluate the impact of adopting a Bi-LSTM design against a unidirectional LSTM architecture. Investigate if the Bi-LSTM's bidirectional nature improves summarization quality and performance indicators such as ROUGE scores or human evaluation.
2. **Dataset Availability:** The availability of data is critical for training and assessing models. Investigate whether there are limits in terms of quantity, quality, or domain coverage in the Assamese dataset utilised in the present study work. We may look towards creating a larger and more diversified dataset for Assamese abstractive text summarization. This might include gathering pertinent news items, blog posts, or other textual sources and annotating them with summaries.
3. **Evaluation Metrics:** The evaluation of the quality of summarization is a key part of this research. Traditional assessment measures such as ROUGE (Recall-Oriented Understudy for Gisting Assessment) may have been applied in the existing research work. These measures, however, have limits, and it would be beneficial to investigate new assessment techniques particular to the Assamese language. Investigate linguistic-based criteria or modify current ones to improve the evaluation of abstractive summarization in Assamese.
4. **Incorporating Language-Specific Features:** Investigate the Assamese language's linguistic characteristics and uncover language-specific factors that might improve the abstractive text summarising process. Assamese, for example, may have distinct grammatical structures, phrase constructions, or speech patterns that might be used in the summary model. Look at including these language-specific variables in our Bi-LSTM model and evaluating their influence on summarization quality.
5. **Comparison with Extractive Methods:** While abstractive summarizing tries to provide human-like succinct summaries, extractive approaches that pick and reorganise existing phrases are an alternate option. Compare your Bi-LSTM-based abstractive summarizing strategy to extractive approaches built expressly for the Assamese language. Compare the two techniques' quality, coherence, and overall performance to determine their respective strengths and limitations.

2 Methodology

This section will provide examples of how we went about creating an abstract Assamese text summarizer. We aimed to develop a text summarizer that can provide an accurate summary of a given text because there haven't been many attempts to summarize Assamese writings. This model was created and trained using Tensorflow CPU version 1.15.4. Figure 1 depicts the method used in our model.

2.1 Dataset Creation

Text summarization typically makes use of standard datasets. In order to create a standard dataset, we followed the conceptual framework provided by Hermann et al.[11] In addition, we looked at some of the public English datasets that are currently accessible, such as the CNN-Daily Mail4 collection. We require a large amount of data for training, however, there isn't a sizable standard dataset readily accessible for Assamese summarization. So, using asomiyapratidin[12], we were able to compile a dataset with over 7000 articles and their related human-written summaries. Online news contains a lot of adverts, non-Assamese language, connections to other websites, and other nonsense. As a result, we began pre-processing by creating a programme for data cleaning that removes any traces of clutter from the dataset.

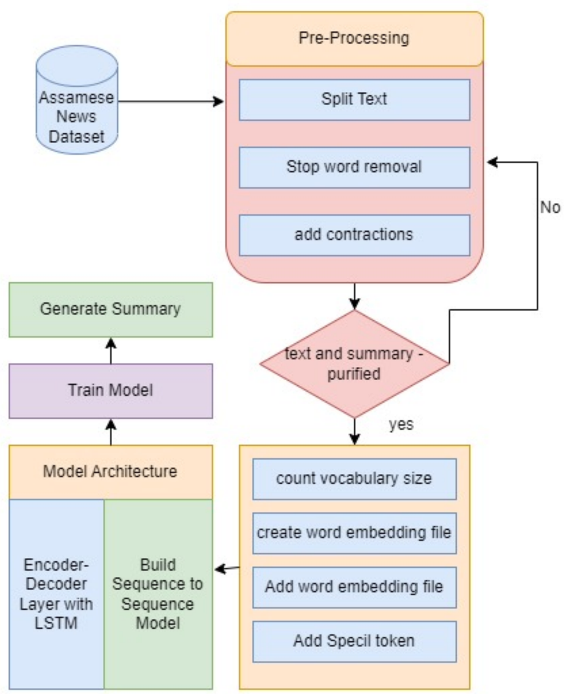

Fig 1. Work flow diagram

2.2 Data Pre-processing

We have completed a few stages in the pre-processing of the data. We have initially added contractions to the text descriptions and summaries. After that, the texts were cleaned. That means all superfluous characters have been removed. These unnecessary parts have been removed from the texts using regular expression. After that, we got rid of stop words[13].

2.3 Split text

After being uploaded, the full dataset needed to be tokenized. The big sentence was reduced to a single word. It helps to tidy up the text and remove unnecessary functions from the dataset. Additionally, it helps the dataset's vocabulary grow, which is important for NLP problems. This vocabulary may be used to extract important information from word embedding files.

2.4 Add contractions

Every language has word contractions. Furthermore, Assamese doesn't use a lot of word contractions. It alludes to the shorthand or abbreviation of a term. Since the computer cannot read texts in their short versions, it is required to explain the full meaning of a term in its short form [14].

2.5 Regular expression

The regular expression is used to delete strange or unwanted characters from the text. In our study, a regular expression is mostly utilized to eliminate spaces, punctuation, Assamese numbers, English letters, and whitespace from the text.

2.6 Stop word remove

Stop words are frequently eliminated from text using NLP techniques. Stop words are mostly used to remove superfluous words from manuscripts. English texts sans stop words may be created using the NLTK built-in library. There isn't a library nearby, though; where Assamese stop words may be removed. Therefore, we begin by compiling all Bengali stop words from online resources. Totalling 226 stop words, we put them into a file for potential use.

2.7 Purified text and summary

Once the aforementioned steps are complete, the text and summary will appear to be in a logical arrangement. In this entire text or summary, there is no punctuation or extra space. Every word appears in a certain order. The clean text and the summary are both added to two different lists. The sequence of these lists serves as the input to the summarization model.

2.8 Pre-processing for Seq2Seq model

The terminology now includes several novel tokens as <UNK>, <PAD>, <EOS>, and <GO>. The vocabulary is subject to a number of restrictions. For missing vocabulary words, use the symbol <UNK>. A <PAD> token adds a batch of sentences, each of which has the same length. The <EOS> character designates the end of the sequence for input to the encoder. The decoder's output sequence operation is initiated with the <GO> token. Before deciding on <EOS> and <GO>, we did sequence translation to the data that contains words. The mode x of this sequence corresponds to the encoder's input, and the mode y to the generated output or response output [15].

2.9 Word embedding

Word similarity affects meaning just as much as word frequency. As a result, we must determine the total number of words in the revised text's descriptions and summary. We transformed a word text into a vector file that we utilised in our model using data from our own dataset. The Gensim Word2vec Skipgram model was trained just on dataset with the following measures: size=300, window size=10, and min count=1 to produce 300-dimensional Assamese word vectors [16].

2.10 Model

- **RNN Encoder and Decoder**

After the development of machine translation, a deep learning approach creates an important milestone in the field of artificial intelligence. All text-related difficulties are reliably addressed by the deep learning model. RNN is the deep learning algorithm that is most useful. Any text-related difficulty is handled better by it. A single LSTM cell forms the basis of each RNN. An LSTM cell is comparable to short-term memory. Encoders and decoders are used in an LSTM cell. As the input text is sent through the encoder, each input is a word vector sequence. The decoder uses the input sequence to generate the text output from the relevant text sequence.

RNNs are available in two different configurations: directional and bidirectional. An input and an output are present in a single-directional RNN, and they are progressively connected to one another. The bi-directional RNN consists of two layers with two orientations. There are two; one advance while the other retreats. All of those are used to remedy the machine translation issue. In our study, we used a two-layered RNN. Since we used RNN for Assamese, the model's encoder input is fixed Assamese text, which the model may carry out. The decoder gives the output's corresponding sequence in accordance with the input.

The first two stages of the RNN encoder-decoder architecture are presented by Cho et al. [17]. Later, this was made worse by Bahdanau et al. [7]. These encoder and decoder types were exclusively used for machine translation.

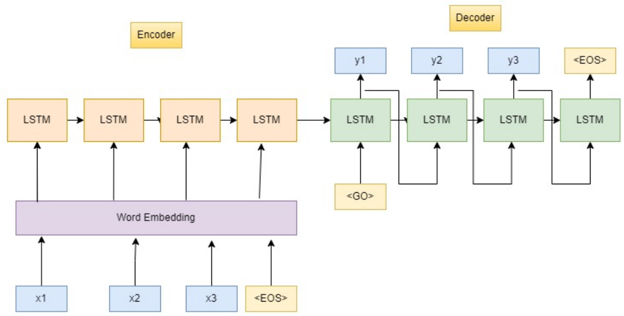

Fig 2. View of Seq2Seq model.

This RNN-equipped, two-layer neural network was used. The encoder holds the fixed length of a phrase, and the decoder holds the output sequence. To retain the maximum conditional probability for the intended text sequence, the two layers of the RNN network are trained concurrently. Memory training and capacity were increased by using a concealed unit. We train our model to determine the probability that an Assamese sentence will be followed by a sentence that is similar to it.

When a target is read by an encoder, enter a sentence X=[x_1,....x_{Tx}] where Table 1 and Table 2 words are used as the model's input and context vector c exists, so that [24]

$$h_{t=} h_{t-1} \tag{1}$$

And

$$c = q\left(h_{1.........}h_{T_x}\right) \tag{2}$$

Where c stands for the context vector produced from the sequence of hidden states and stands for the hidden state at time t, f and g are the nonlinear functions.

If the word sequence predicted by the decoder and the return summary of Table 1 and Table 2 match

$$p\left(y\right) = \prod_{t=1}^{T} p\left(y|\{y_{1,.........,}y_{t-1}\}\right), (c) \tag{3}$$

Where, $[y_1,.....,y_{T_0}]$ Conditional probability is represented by the model shown below:

$$p(y|\{y_1,.......,y_{t-1}\},c) = g(y_{t-1},s_t,c) \tag{4}$$

Here y_t is a probability outcome, $s_t$is a hidden state, whereas g is a non-linear function.

$$c = \sum_{j=0}^{T} a_{ij}h_j \tag{5}$$

In this case, bi-directional recurrent neural networks, which comprise recurrent neural networks that move both forward and backward, were used.

A forward RNN's hidden state is $[\overrightarrow{h_1},......\overrightarrow{h_{T_x}}]$ and its sequence order is $x_1 \ to \ x_{T_x}$. A backward RNN's hidden state is $[h_1,......,h_{T_x}]$, and its sequence order is $x_{T_x} \ to \ x_1$. Therefore,

$$h_j = \left[\overrightarrow{h_{T_x}};\overset{\rightharpoonup}{h_{T_x}}\right]^T \tag{6}$$

Where h_j is a summary of the terms that were anticipated and used.

Furthermore, a_{ij} is equal to the normalised exceptional function called the softmax which is e_{ij}, that shows how well the input value j and the output value i are matched [15],

$$e_{ij} = a\left(S_{i-1},h_j\right) \tag{7}$$

- **Learning from Sequence to Sequence**

Each encoder and decoder in a sequence-to-sequence model uses an LSTM cell. We have a word embedding file that we have utilised in our text summarize approach. Then, we measured the vocabulary length of such files that served as the input for the model.

We also added some unique vocabulary terms like <UNK>, <PAD>, < EOS> and <GO>. There are various limitations on vocabulary. Some words are still in use. <UNK> token is used in place of the words. Each sentence in a batch added by a <PAD> token is the same length. The end of the sequence that alerts the encoder when it receives input is contained within the <EOS> token. The <GO> token gives the decoder the command to begin processing the output sequence. We rebuild the vocabulary and add <UNK> during the data preparation stage. We applied sequence translation on the data that contains words before choosing <GO> and <EOS>. In this, the sequence for mode y is the produced output sequence, while mode x is the encoder's input sequence [15].

3 Results and Discussion

The techniques outlined in the preceding section are used in this study to extract a virtually correct summary from news articles. This model was trained over a period of more than 3 hours using the standard training settings of 150 epochs and a 2-batch size. It reduced the training loss by 0.008 percent on average. After that, a TensorFlow session running version 1.15.2 loads the previously stored graph once more. The data frames for the news and summary are also produced at random. To be exact, this is an input-output system. This methodology yields findings that are essentially same for most articles. It does, however, occasionally provide summaries that are less accurate than the originals. Table 1 present illustrations of results from this dataset.

Table 1. Sample result 1

Original Text:	আন্তঃৰাষ্ট্ৰীয় পৰ্যায়ত উজ্বলিলভাৰতৰ খেলুৱৈ।টাইৱানৰটাইপে'ত অনুষ্ঠিত দ্বিতীয়ৱৰ্ল্ডডীফ য়ুথ বেডমিন্টনচেম্পিয়নশ্বিপত সোণৰপদক দখল জেৰ্লিন অনিকাৰ।মাদুৰাইৰ ১৫ বছৰীয়া খেলুৱৈগৰাকীয়েবেডমিন্টনত সোণ অৰ্জন।খেলুৱৈগৰাকীৰমাতৃয়ে কয় 'জেৰ্লিনৰ সফলতাতমই অত্যন্ত সুখী।''এই যুৱখেলুৱৈগৰাকীক সহায়ৰমই চৰকাৰক সহায়ৰহাত আগবঢ়াবলৈ অনুৰোধজনাইছো।''জেৰ্লিনে অৰ্জনসোণৰ পদক তাইৰনহয় এয়া সমগ্ৰদেশৰ হয়।'
Original Summary	ৱৰ্ল্ডডীফ য়ুথবেডমিন্টন চেম্পিয়নশ্বিপতভাৰতৰ খেলুৱৈৰ সোণৰপদক অৰ্জন
Input Words	আন্ত ৰাষ্ট্ৰীয় পৰ্যায়ত উজ্বলিল ভাৰতৰ খেলুৱৈ।টাইৱানৰ টাইপেত অনুষ্ঠিত দ্বিতীয় ৱৰ্ল্ডডীফ য়ুথ বেডমিন্টন চেম্পিয়নশ্বিপত সোণৰ পদক দখল জেৰ্লিন অনিকাৰ।মাদুৰাইৰ বছৰীয়া খেলুৱৈগৰাকীয়ে বেডমিন্টনত সোণ অৰ্জন খেলুৱৈগৰাকীৰ মাতৃয়ে কয় জেৰ্লিনৰ সফলতাত মই অত্যন্ত সুখী এই যুৱ খেলুৱৈগৰাকীক সহায়ৰ মই চৰকাৰক সহায়ৰ হাত আগবঢ়াবলৈ অনুৰোধ জনাইছো জেৰ্লিনে অৰ্জন সোণৰ পদক তাইৰ নহয় এয়া সমগ্ৰ দেশৰ হয়
Response word	ৱৰ্ল্ডডীফ য়ুথ বেডমিন্টন চেম্পিয়নশ্বিপত ভাৰতৰ খেলুৱৈৰ সোণৰ পদক

We used beam search of size 10 to build the summary at decode-time and put the maximum word count at 30, given that the longest summary in the dataset is 62 words. We give Precision, Recall, and F1-scores from the full-length versions of Rouge-1, Rouge-2, and Rouge-L using the official evaluation script. We evaluate our system using Rouge's full-length F1 form, following Nallapati et al. and Chopra et al. Restrictions must be followed while determining the length, which varies from corpus to corpus and makes it challenging for researchers to compare data. Restricted length recall is another common statistic. On the other hand, full-length recall does not impose a limit on the length but unfairly favours longer summaries. Full-length F1, which does not enforce a specific summary length limit but can penalise longer summaries, resolves this issue. Figure 3's bar diagram displays the model's ROUGE scores. We also compare our findings with some abstractive text summarization in Bengali, and its results are fairly excellent since Bengali language structure is quite close to Assamese.

Deep learning approaches, particularly sequence-to-sequence (Seq2Seq) models incorporating attention mechanisms, have shown substantial promise for abstractive text summarization in a variety of languages. However, study in this field is presently restricted to Assamese, making it difficult to compare our findings to those of others. Nonetheless, with the exception of the letters '"ক্ষ (Khyo)" and "ৰ (ro),"Bengali, commonly known as Bangla, is the language most similar to Assamese, with the bulk of their linguistic traits being identical. As a result, we conducted a comparison of our work with abstractive text summary in Bengali [18].

Chowdhury et al. abstractively summarized 139 human-written document-summary pairs generated by experienced summary writers at the National Curriculum and Textbook Board (NCTB) [19]. In our study, we used 7000 texts and their related human-generated summaries. Our ROUGE assessment scores outperform theirs, as seen in Table 2 below.

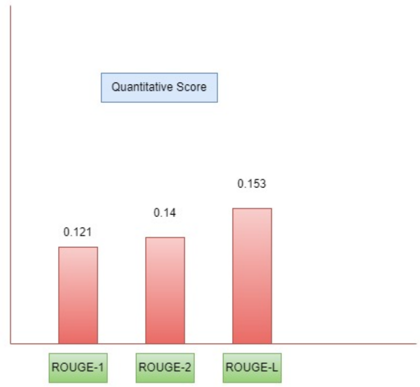

Fig 3. Model's ROUGE scores

Table 2. Comparison with previous work

	Dataset	ROUGE-1	ROUGE-2	ROUGE-L
[34]	139 text document	12.17	NA	11.35
Proposed work	7000 text document	12.10	14.00	15.30

4 Conclusion

One of the most successful applications of seq2seq models, neural abstractive text summarization, has grown into a significant research field that has piqued the interest of both industry and academia. In this study, we try to employ a deep learning architecture called a sequence-to-sequence encoder-decoder to address the issue of abstractive text summarization for the low-resource Indo-Aryan language Assamese. We also evaluated the summary generation using the well-known ROUGE metric. We also looked at the good and bad abstractive summaries that were created. To the best of our knowledge, this is the first effort of its sort where a deep learning model has been effectively deployed for abstractive summarization in Assamese on an Assamese news dataset. In the future, we would like to use cutting edge BERT models for abstractive text summarization or pointer-based copying mechanisms to solve the repetition management problem.

5 Declaration

Presented in Fourth Industrial Revolution and Higher Education (FIRHE 2023) during 23rd-25th Feb 2023, organized by DUIET, Dibrugarh University, India. The Organizers claim the peer review responsibility.

References

1) Abate J, Tulu A. Development of a Query-Based Document Summarization for Afaan Oromoo. . Available from: https://gu-clasp.github.io/resourceful-2020/papers/RESOURCEFUL-2020_paper_1.pdf.
2) Jain R. Automatic Text Summarization of Hindi Text Using Extractive Approach. *ECS Transactions*. 2022;107(1). Available from: https://doi.org/10.1149/10701.4469ecst.
3) Moirangthem DS, Lee M. Abstractive summarization of long texts by representing multiple compositionalities with temporal hierarchical pointer generator network. *Neural Networks*. 2020;124:1–11. Available from: https://doi.org/10.1016/j.neunet.2019.12.022.
4) Baruah R, Mundotiya RK, Singh AK. Low Resource Neural Machine Translation: Assamese to/from Other Indo-Aryan (Indic) Languages. *ACM Transactions on Asian and Low-Resource Language Information Processing*. 2021;21(1):1–32. Available from: https://doi.org/10.1145/3469721.
5) Varade S, Sayyed E, Nagtode V, Shinde S. Text Summarization using Extractive and Abstractive Methods. *ITM Web of Conferences*. 2021;40:1–5. Available from: https://doi.org/10.1051/itmconf/20214003023.
6) Bahare RS, Ganguly S, Choowongkomon K, and SS. Synthesis, HIV-1 RT inhibitory, antibacterial, antifungal and binding mode studies of some novel N-substituted 5-benzylidine-2, 4-thiazolidinediones. *DARU Journal of Pharmaceutical Sciences*. 2015;23(6):1–15. Available from: https://doi.org/10.1186/s40199-014-0086-1.
7) Bahdanau D, Cho K, Bengio Y. Neural Machine Translation by Jointly Learning to Align and Translate. 2014. Available from: https://arxiv.org/abs/1409.0473.
8) Yang Z, Liu G. Hierarchical Sequence-to-Sequence Model for Multi-Label Text Classification. *IEEE Access*. 2019;7:153012–153020. Available from: https://doi.org/10.1109/access.2019.2948855.

9) Uteshov Y. Automatic Text Summarization Using Sequence-To-Sequence Models. *Internauka*. 2022;19(242). Available from: https://doi.org/10.32743/26870142.2022.19.242.339376.
10) Babu GLA, Badugu S. Deep learning based sequence to sequence model for abstractive telugu text summarization. *Multimedia Tools and Applications*. 2023;82:17075–17096. Available from: https://doi.org/10.1007/s11042-022-14099-x.
11) Liu PJ, Saleh M, Pot E, Goodrich B, Sepassi R, Kaiser L, et al. Generating Wikipedia by Summarizing Long Sequences. 2018. Available from: https://arxiv.org/abs/1801.10198.
12) Uddin MN, Khan SA. A study on text summarization techniques and implement few of them for Bangla language. In: 2007 10th international conference on computer and information technology, 27-29 December 2007, Dhaka, Bangladesh. IEEE. 2007. Available from: https://ieeexplore.ieee.org/abstract/document/4579374.
13) Nath P, Adhikary PK, Dadure P, Pakray P, Manna R, Bandyopadhyay S. Image Caption Generation for Low-Resource Assamese Language. In: The 34th Conference on Computational Linguistics and Speech Processing (ROCLING 2022), November 21-22, 2022, Taipei, Taiwan. 2022;p. 263–272. Available from: https://aclanthology.org/2022.rocling-1.33.pdf.
14) Baruah R, Mundotiya RK, Singh AK. Low Resource Neural Machine Translation: Assamese to/from Other Indo-Aryan (Indic) Languages. *ACM Transactions on Asian and Low-Resource Language Information Processing*. 2021;21(1):1–32. Available from: https://doi.org/10.1145/3469721.
15) Talukder MAI, Abujar S, Masum AKM, Faisal F, Hossain SA. Bengali abstractive text summarization using sequence to sequence RNNs. In: 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 06-08 July 2019, Kanpur, India. IEEE. 2019. Available from: https://doi.org/10.1109/ICCCNT45670.2019.8944839.
16) Yang J, You J, Wan X. Graph Embedding via Graph Summarization. *IEEE Access*. 2021;9:45163–45174. Available from: https://doi.org/10.1109/access.2021.3067901.
17) Cho K, Van Merrienboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, et al. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. Association for Computational Linguistics. 2014. Available from: https://arxiv.org/pdf/1406.1078.pdf.
18) Goutom PJ, Baruah N, Sonowal P. An abstractive text summarization using deep learning in Assamese. *International Journal of Information Technology*. 2023;15:2365–2372. Available from: https://link.springer.com/article/10.1007/s41870-023-01279-7.
19) Chowdhury RR, Nayeem MT, Mim TT, Chowdhury MSR, Jannat T. Unsupervised Abstractive Summarization of Bengali Text Documents. 2021. Available from: https://arxiv.org/pdf/2102.04490.pdf.