RAYA JOURNAL OF SCIENCE AND DEVELOPMENT
RJSD 1(1):2025

Abstractive Tigrigna Text Summarization using Deep Learning Approach

Yemane Gebrekidan [1], Gebrekiros Gebreyesus [2], Tewelde Hailay [1], Yemane Hailay Nega[1]

[1] Department of Computer science Raya University, Maichew, Ethiopia

[2]Department of Electrical Engineering Raya University, Maichew, Ethiopia

Article History:
Received: 2025-08-05
Accepted: 2025-09-24
Published: 2025-11-06
DOI: https://doi.org/10.82489/rjsd.2025.1.01.29

Suggested citation:
Yemane, G., Gebrekiros, G., Tewelde, H., & Yemane, H. (2025): Abstractive Tigrigna Text Summarization using Deep Learning Approach. RJSD 1(1): 1 -14; DOI: 10.82489/rjsd.2025.1.01.29.

Correspondence :
Yemane Gebrekidan Gebretekle
yemanegkidan8@gmail.com

Abstract

Text summarization is becoming an essential research area due to rapid growth of online text data. It involves condensing long sequences of text into short, concise, and expressive summaries. There are two main approaches: extractive and abstractive. Extractive approach focuses on selecting portions of text without addressing underlying meaning. While abstractive approach rephrases or reorganizes long text to produce semantically equivalent summaries, possibly with new words or phrases. In the case of Tigrigna, most existing studies have relied on extractive techniques, leaving abstractive methods largely unexplored. There is no structured dataset, no pre-trained word embedding, and no pertained summarization model in Tigrigna. These fundamental problems have been affecting the motive to apply abstractive approach in Tigrigna. This study addresses this gap by developing an abstractive text summarization model for the Tigrigna language using deep learning techniques. A dataset of 1,167 structured input paragraphs and reference summaries was prepared for training and evaluation. Different embedding methods, including fastText and Byte Pair Encoding, were trained on about 320 MB of data. In this study, two models (Sequence-to-sequence Long Short-Term Memory and Transformer) were evaluated. The Sequence-to-sequence works sequentially, whereas the Transformer operates in parallel. An attention mechanism was added to Sequence-to-sequence, while Transformer uses self-attention. Among tested model–embedding matches, Sequence-to-sequence with attention and fastText with down-sampling showed superior performance, achieving accuracy of 0.72 and Recall-Oriented Understudy for Gisting Evaluation scores of R-1=0.20, R-2=0.183, and R-N=0.17. This work pioneers Tigrigna abstractive summarization, marking a foundational step for future research. Future studies could concentrate on growing the dataset, investigating bidirectional and hybrid deep learning architectures.

Keywords: *abstractive, attention mechanism , fast text embedding, LSTM, Seq2Seq*

1. Introduction

Natural Language Processing (NLP) has been transforming how machines interpret and process written text (Regulation, 2024). Text summarization has emerged as an essential NLP task for generating concise and meaningful summaries with the exponential growth of online content. Among the many languages processed by NLP systems, Tigrigna, a morphologically rich language spoken by around 9.9 million people in the Horn of Africa, faces significant challenges due to the scarcity of resources such as annotated datasets, pretrained embeddings, and summarization models (Birhanu, Guesh Amiha, 2017). This lack of tools has slowed research progress and limited information accessibility for Tigrigna speakers.

Text summarization can be broadly categorized into extractive and abstractive approaches (Shakil et al., 2024). Extractive summarization selects key portions of the original text, while abstractive summarization generates new sentences that capture the overall meaning(Relan & Rambola, 2022). For Tigrigna, existing studies have focused mainly on extractive methods, which often fail to preserve coherence and semantic richness(Carenini et al., 2006).

This study, therefore, explores the application of modern abstractive text summarization techniques to Tigrigna. It focuses on methods based on the Sequence-to-Sequence (Seq2Seq) framework using the encoder-decoder model (Bo et al., 2025; Wazery et al., 2022). In this model, the encoder transforms input text into hidden representations, while the decoder generates a new sequence of tokens as output. The main objective of this research is to apply and adapt NLP frameworks to develop an abstractive summarization system for Tigrigna. The study involves preparing datasets and embeddings, building deep learning models such as Seq2Seq with attention and Transformer architectures, and evaluating their performance.

2. Literature Review

Text summarization is an active study topic in today's world. As a result, there are numerous studies underway in this area. To that purpose, the researcher presents works from the beginning to the present, demonstrating how different researchers conducted their research and what they learned. Based on the output they generate; text summarization approaches are either extractive or abstractive(Carenini et al., 2006).

Some studies to Tigrigna extractive summarization of single documents are researched(Birhanu, Guesh Amiha, 2017; Birhanu, 2017; Regassa et al., 2017). These researches focused on identifying key textual elements. For example, proposed a topic-based Tigrigna text summarization method that integrates wordnet and Probabilistic Latent Semantic Analysis (PLSA). This research was evaluated using a dataset of 200 Tigrigna news articles from various domains. The performance

was assessed with precision and recall metrics at a 25% extraction rate, yielding a precision/recall score of 0.5014.

A review has also done on many papers using abstractive approach. Though there is not any abstractive based paper published for Tigrigna language, the researcher reviewed some related languages like Amharic, English, Urdu and Arabic(Khalil, 2020; Shafiq et al., 2023; Tamiru & Libsie, 2009; Yirdaw, 2011; Yirdaw & Ejigu, 2012). These papers have applied deep learning to train their models. Deep learning is a trending field due to its essential applications in the research area(Abdullahi et al., 2021). There are various models that suits, the abstractive text summarization(Paritosh Marathe, 2020). There are even pre-trained or predefined models to the popular languages like English. However, for the Ethiopian languages, it is not exercised well. Particularly in Tigrigna, there is no pre-trained deep learning model.

3. Research Methodology

The setting of the methodology for the abstractive Tigrigna text summarization includes data collection and structuring, data preprocessing, word embedding, padding, and model implementation and training.

Table 1

Summary of Related Works in Tabular Form

Title	Language	Author and Year	Method
Topic-Based Text Summary	Tigrigna	Regassa & Getachew, 2017;	Wordnet and PLSA
Automatic Text Summarizer	Tigrigna	Birhanu, 2017	Term frequency and title words methods
Text Summarization Using Deep Learning	Tigrigna	Hiluf Gebrehiwot and Melese, 2023	Restricted Boltzmann Machines (RBM)
Abstractive Text Summarization Model	Amharic	Khalil, 2020	Seq2Seq with Long Short-Term Memory (LSTM) and attention
Deep Learning-Based Abstractive Summarization	English	Suleiman & Awajan, 2020	RNN with attention and LSTM
Deep Learning for Abstractive Summarization	Urdu	Shafiq et al., 2023	Seq2Seq model with deep learning compared to SVM and LR
Text Summarization with Seq2Seq and Attention Mechanism	English	Sutskever et al., 2014	Seq2Seq with attentional encoder-decoder RNN

3.1 Dataset Structuring

Data collection and structuring tasks were the most challenging steps due to the scarcity of prepared resources in Tigrigna language. The data is taken from Dimtsi Weyane television, Tigrai television, and Github website(*Tigrinya · GitHub Topics · GitHub*, n.d.), as raw data. These environments were chosen in two reasons for data collection. Firstly, it is easy to collect the desired data. The second and mandatory criteria was the nature of their data. This means most of the data are news texts. Usually, news is prepared by journalists, having the main news (title) so that the title is used as the reference summary. Therefore, 1,167 pairs of input texts

(paragraphs) and reference summaries are structured by using the title part as the reference summary to help the model learn summarization in Tigrigna.

While working in abstractive summarization using deep-learning, it is essential to structure and organize the data before preprocessing (Abdullahi et al., 2021). In this case, the model needs pair of input text and reference summary, as input for the encoder-decoder LSTM network. And hence these pair of inputs are prepared in notepad editor by adding special delimiters '===text===' (user defined delimiter for the Text) to the input text and '===summary===' (user defined delimiter for the Summary) to the reference summary. These delimiters are used to identify the pair of inputs while loading the dataset file, and those delimiters are custom markers that defined to separate and identify segments of data when preparing input files manually for preprocessing scripts. The following Tigrigna texts show how the given Tigrigna text is structured before preprocessing.

Raw text example:

ኣርእስተ ዜና

ኣብ ትግራይ ምስፍሕፋሕ ኢንቨስትመንት ንሆቴላት ምችው ባይታ ዝፈጥር እዩ ተባሂሉ

ዝርዝር ዜና

ምስፍሕፋሕ ኢንቨስትመንት ሆቴላት ዕድል ስራሕ ኣብ ምፍጣርን፣ ዳግመ ህንፀትን መስሕብ ቱሪዝምን ትግራይ እውን ኣብ ምብራኽን ልዑል ኣበርክቶ ከምዘለዎም ቢሮ ባህልን ቱሪዝምን ትግራይ ገሊፁ። እዚ ዝተገለፀ ብልዕሊ 200ሚልዮን ብር ወፃኢ ኣብ ከተማ መቐለ ዝተሃነፀ ሆቴል ከይከይ (KK) ኣብዝተመረቐሉ እዋን እዩ።

The following text shows input pairs of the above news after structuring:

===text=== ምስፍሕፋሕ ኢንቨስትመንት ሆቴላት ዕድል ስራሕ ኣብ ምፍጣርን፣ ዳግመ ህንፀትን መስሕብ ቱሪዝምን ትግራይ እውን ኣብ ምብራኽን ልዑል ኣበርክቶ ከምዘለዎም ቢሮ ባህልን ቱሪዝምን ትግራይ ገሊፁ። እዚ ዝተገለፀ ብልዕሊ 200ሚልዮን ብር ወፃኢ ኣብ ከተማ መቐለ ዝተሃነፀ ሆቴል ከይከይ (KK) ኣብዝተመረቐሉ እዋን እዩ።

===summary=== ኣብ ትግራይ ምስፍሕፋሕ ኢንቨስትመንት ንሆቴላት ምችው ባይታ ዝፈጥር እዩ ተባሂሉ::

3.2. Data Preprocessing

Data preprocessing involves preparing raw text data to enhance the performance of summarization models. This task includes text cleaning, contraction mapping, character normalization, and tokenization. Text cleaning also involves removing non-Tigrigna characters, punctuations, special characters, stop words (like ስለ፣ ከም) and extra whitespaces. Contraction mapping and character normalization is also done on the Tigrigna contraction words and some characters that have same sound and purpose. In this case, a separate JSON file is prepared as input for the preprocessing. This file includes Tigrigna characters mapping (e.g., ጸ=ፀ፣ ሠ=ሰ) and Tigrigna words abbreviation mapping (e.g., ዝ/ዩ=ዝተፈላለዩ፣ ቤ/ት/ቲ=ቤት ትምህርቲ). Tokenizing the text into words or sentences, and normalizing words through lemmatization or stemming. This file is prepared manually from Tigrai tourism and culture bureau. It has 92 abbreviations and short form mappings, and 28-character mappings.

Additionally, tokenization is also done to the cleaned data. This activity reads a sequence of characters as a string and tokenizes them using a predefined list of delimiters such as space and

punctuation marks and then represents each token in array of numbers. In this study, input texts are tokenized into a sentence using the punctuation "።" and each sentence are spliced in to entire or nested list which helps to convert into list of words.

3.3. Data Splitting

Data splitting is one of the mandatory steps in preparing data to train deep learning models. It is the process of intentional dividing the organized data in to different subsets to be used in different stages of the model's lifecycle such as training, validation, and testing.

In this work, data is split using a ratio of 80% for training, 10 % for validation, and 10% for testing subsets. This ratio of the subsets is selected by its convenience to the model and data size after conducting 3 experiments with common data splitting ratios in abstractive text summarizations. The dataset is limited and was partitioned with 10% reserved for testing. Without allocating a portion of the data for validation and testing, the model would be susceptible to overfitting, compromising its generalization performance. There are different techniques to split the data in to different subsets. The commonly exercised techniques are random and stratified. Random method is used for splitting as the data is uniformly distributed.

3.4. Word Embedding

Word embeddings help models understand the meaning of words in context(Egger, 2022). For Tigrigna, a language with rich morphology and syntactic structures, embeddings need to capture not only word meanings but also relationships between different word forms. Tigrigna, like many other languages, has words that change form based on tense, number, and gender. Specialized embeddings can capture these variations, allowing the model to generalize better and produce more accurate summaries.

There are two types of word embedding techniques: traditional methods and contextual methods(Egger, 2022). In the traditional types of embedding techniques, the vector form does not capture the semantic relationship as words are embedded independently. Examples of traditional word embedding include Term Frequency-Inverse Document Frequency (TF-IDF), count vector, GLOVE, Word2Vec etc (Abdullahi et al., 2021). On the other way, the common algorithms used for training word embeddings having semantic information are fastText, Elmo, Generative Pre-trained Transformer (GPT) and GPT-2, Byte Pairing Encoding (BPE) and Bidirectional Encoder Representations from Transformers (BERT). In this study, experiments for word embedding were made using fastText and BPE word embedding methods for their convenience to rich morphology languages(Egger, 2022). Due to the absence of publicly available pre-trained Tigrigna word embeddings, custom models were developed for this research. A custom Tigrigna word embeddings were prepared using those approaches and trained them on a large dataset

spanning multiple fields like agriculture, education, sports, and politics. As a consequence of the experiment, the fastText using down sampling method has performed better and used to generate custom embedding of the data set. The resulting word vectors are maintained locally, providing an important linguistic resource for the research goal, developing an abstractive summarization model for Tigrigna text.

In this technique, every single occurrence of a word is represented by embedding dimension. Embedding dimension defines the length of the vector representation for each word. Higher value reflects higher information but requires high memory space. The common values are 50, 100, 200, and 300. When the data is large size, using large dimension (e.g. 300) is preferred. However, when data is small, it is recommended to set average dimension (e.g. 50 or 100) to utilize memory space(Egger, 2022). In this work, the embedding dimension is set to 100 as you can see in the output example below.

Example: the word 'ትምህርቲ' is represented as follows.

ትምህርቲ 0.1925859 -0.29207048 -0.08439901 0.039799646 0.3641676 0.073608406 -0.7650273 0.04147037 0.2879626 -0.36144435 -0.39172873 0.5143982 -0.752826 0.08606512 -0.93490994 -0.13160115 -0.41230726 0.7842884 0.5688561 0.10861172 -0.26581797 -0.23717366 0.25146484 -0.7285605 -0.9428908 -0.015943918 1.1289601 -0.7270743 -0.53325266 0.045723584 -0.4794562 0.94648546 -0.5338172 -1.165367 0.2542716 -0.52565515 -0.3413985 -0.03239215 -0.5524072 -0.10506536 0.062022235 0.8638829 -1.0330185 0.7314383 0.51415807 0.33765644 -0.17070593 0.032447204 -0.2572221 -0.0751654 -0.5847026 -0.6834384 -0.4850279 0.09068502 -0.6141403 0.45645028 0.16440077 -0.47739515 0.35700983 -0.47724828 -0.08733774 0.29962227 0.38471565 -0.038393848 -0.88605523 0.17655876 0.027320378 0.756702 -0.35022673 0.42926 -0.118354924 0.35008702 0.5892428 -0.27457365 -0.8334408 -0.023459285 -0.1667422 0.92475754 0.04237065 0.13801275 0.6401937 -0.22754209 0.18134119 -0.3616498 0.69549537 0.86129206 0.039201967 -0.06344785 0.1806198 0.5076103 0.6642553 -0.49786687 0.44641012 0.24511304 -0.09361779 -0.9122355 -0.1879493 0.01642124 -0.852965 -0.07018303

3.5. Proposed Deep Learning Models

Deep learning can automatically learn complex linguistic patterns. Hence, it has emerged as a dominant paradigm in NLP (Abdullahi et al., 2021). The researcher developed and trained two deep learning models for the current study, which focus on abstractive text summarization in Tigrigna: Transformer model and Seq2Seq model with LSTM and attention mechanism. These models were chosen following an analysis of a number of advanced architectures and their applicability to low-resource, morphologically rich languages such as Tigrigna. Since no pre-trained summarization models exist for Tigrigna, all models were implemented and trained from scratch to enable fair performance comparison of the proposed approaches.

3.5.1. Seq2Seq Model with LSTM and Attention Mechanism

The first proposed model is a Seq2Seq architecture composed of two core components: an encoder and a decoder, both implemented with LSTM networks.

Encoder: the encoder LSTM takes the input text sequence (Tigrigna sentences) and encodes it into a fixed-length context vector representing the semantic meaning of the entire sentence.

Decoder: the decoder LSTM receives both the context vector and the attention weights to generate the target summary word by word.

Attention layer: instead of relying solely on the final encoder state, the model integrates an

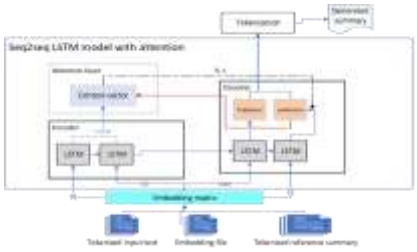
attention mechanism that allows the decoder to selectively focus on different parts of the input sequence during the generation process. This is crucial for Tigrigna, which often uses long and morphologically complex words.

This architecture helps the model overcome the limitations of basic Seq2Seq models that struggle with long-range dependencies. In training phase, the model was optimized using categorical cross-entropy loss and the Adam optimizer, with early stopping applied to prevent overfitting.

In **Figure 1**, the detail architecture of the seq2seq LSTM model with attention mechanism is presented. It shows how the model processes the given sequence of text and generates abstract using the deep learning algorithms. The LSTM networks are selected in their advantages like handling long term dependencies, memory cell and forgetting mechanism, and generally suitability for sequential data, over the other traditional recurrent neural networks.

Figure 1

Text of Specified Style in Document.1 Architecture of Seq2seq LSTM Model with Attention Layer



3.5.2. Transformer Model

The second proposed model is a Transformer-based architecture, which relies entirely on self-attention instead of recurrent layers (Bo et al., 2025). The Transformer consists of stacked encoder and decoder layers. Each encoder layer includes multi-head self-attention and feed-forward sublayers, while the decoder includes masked self-attention, encoder–decoder attention, and feed-forward networks. Transformer also have self-attention mechanism. This mechanism allows the model to consider all positions of the input sequence simultaneously, capturing long-range dependencies efficiently. This parallelism also makes the model faster to train compared to LSTM-based architectures. Since Transformers lack recurrence, positional encoding was added to the input embeddings to provide information about word order.

The same preprocessing pipeline was used during training, as the Seq2Seq model was applied. The Transformer demonstrated stronger ability to capture contextual meaning across longer sentences but required larger computational resources.

3.6. Experiments and Hyperparameters Tunning

3.6.1. Hyperparameter Tunning

The seq2seq LSTM model with attention mechanism and fastText embedding is trained under the following settings. Selecting the right hyperparameters is crucial to enhance the model's performance. The basic hyperparameters used in the model training are epoch size, batch size, learning rate, dropout rate, early stopping, optimizer and embedding dimension. These parameters were essential to

manage the performance of the model while training using the data and generating relatively accurate summary text(Abdullahi et al., 2021). Hyperparameters such as epoch and batch size are set manually in the experiment. Too small epoch and batch size is not good for the model learning capability, certainly cause overfitting.

3.6.2. Analysis of Experiment-3

This is the best experiment that shown best result during the model training phases. In this experiment epoch is 30, batch size is 32, and learning rate is dynamically scheduled starting from 0.001. It is possible to make further experiments above epoch 30 or below 20. However, when you go to high number such as 40 or 50, you face numerous challenges such as memory, processor, and mainly model overfitting. There is also the concept of early stopping, that automatically terminates the training if there is no change on the performance metrics. Here, the researcher observed using more than 30 epochs have not positive effect on the model training and hence used maximum epoch value 30.

Table 2

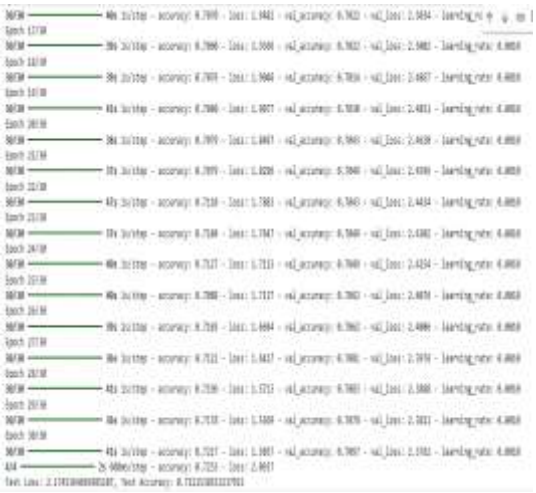
Text of specified style in document.2 Conducted experiments in Seq2seq using LSTM model

Exp	EP	BS	L-rate	P	Factor	loss	Test	Emb
1	*20*	*64*	*0.01*	*5*	*0.2*	*2.196*	*0.718*	*100*
2	*25*	*32*	*0.001*	*5*	*0.2*	*2.206*	*0.714*	*100*
3	*30*	*32*	*0.001*	*5*	*0.2*	*2.174*	*0.721*	*100*

Note. Exp=Experiment, EP= Epoch, BS= Batch-size, P=Patience, and Emb= Embedding

Figure 2

Text of Specified Style in Document.2 Model Training in Experiment-3



In these experiments, it is shown that the hyperparameter tuning and the performance results found in each experiment of hyperparameters. And hence, as shown in **Figure 2** experiment 3 has performed better and taken as the best settings to train the model.

4. Results and Discussion

In this study, the performance results of the different models are reported using accuracy and ROUGE scores across different types of embeddings. During training, accuracy result is generated in every epoch. But the result to be compared across the different models is of the testing data that has reserved as 10 % of the dataset. The Table 3 and 4 below describe the accuracy and ROUGE scores achieved by each model based on the specific embedding methods used. Notably, some models show signs of overfitting which can be attributed to the limited size and domain imbalance of the Tigrigna dataset, as well as the relatively high complexity of the LSTM architecture compared

to the amount of available training data, as evidenced by differences in performance measures between the training and validation periods. To illustrate these insights, the Table 3 compiles accuracy results, while Table 4 provides ROUGE scores for each model and embedding type. This comparison investigation sheds light on how embedding choices affect model performance during Tigrigna summarization. The combination of different model architectures and embedding types yielded varying results. Table 3 below shows the accuracy results of experimented models with varying embedding types**.**

Table 3

Text of Specified Style in Document.3 Accuracy Results

Accuracy results	Using fast Text embedding			Using BPE embedding
	Simple fast Text	Stop words removal	Down-sampling	
Simple Seq2seq LSTM	0.713	0.652	0.715	0.6895
Seq2seq with attention mechanism	0.709	0.700	0.721	0.695
Transformer	0.009	0.22	0.32	0.45

In the area of NLP tasks like text summarization, only accuracy is not sufficient to measure performance. Accuracy cannot capture relevance, coherence, and fluency of generated summaries that summarization quality often depends on. Accuracy generally measures exact matches, which may not fully represent the quality of generated text since a summary can be accurate but still miss key details or be poorly structured.

For summarization tasks, using metrics like ROUGE is crucial (Shakil et al., 2024). ROUGE measures n-gram overlap between generated and reference summaries. In abstractive summarization ROUGE value is usual to be very small size. This is because the new generated summary is expected to be formulated with possibly new words. So, generating new words will decrease the ratio of overlapping n-grams and subsequences. ROUGE includes variations like ROUGE-N (for n-grams), ROUGE-L (for longest common subsequence), and ROUGE-W (weighted overlap). The model has been evaluated by using the three types of ROUGE values, namely ROUGE-N (ROUGE-1 and ROUGE-2) and ROUGE-L. These metrics provide a more nuanced view of how well generated summaries align with human-generated references, beyond mere correctness.

Table 4 shows the different ROUGE results for all proposed models with corresponding to the different embedding types. In this table, the bold values are higher values by average.

Table 4

Text of Specified Style in Document.4 ROUGE Results

ROUGE results	Using fast Text embeddings			BPE embedding
	Simple fast Text	Stop words removal	Down-sampling	
Simple Seq2seq LSTM	**R1=0.20** **R2=0.23** **RL=0.108**	R1=0.13 R2=0.115 RL=0.09	R1=0.197 R2=0.194 RL=0.17	R1=0.12 R2=0.105 RL=0.109
Seq2seq with attention mechanism	R1=0.2 R2=0.171 RL=0.168	R1=0.16 R2=0.135 RL=0.099	**R1=0.205** **R2=0.183** **RL=0.17**	R1=0.16 R2=0.16 RL=0.154
Transformer	R1=0.19 R2=0.15 RL=0.14	R1=0.18 R2=0.18 RL=0.175	R1=0.183 R2=0.174 RL=0.08	**R1=0.20** **R2=0.196** **RL=0.18**

Note. R1 = ROUGE-1, R2 = ROUGE-2, RL = ROUGE-L

Generally, these experiments revealed that both the choice of model architecture and the embedding type, along with specific

preprocessing steps, had a substantial impact on performance. The Seq2Seq with Attention model paired with fast-Text embeddings with down-sampling mechanism yielded the best overall performance. The Transformer model with BPE embeddings also showed encouraging results, particularly for handling complex or lengthy input sequences. These findings provide insight into the factors that influence the effectiveness of models for Tigrigna text summarization and emphasize the importance of appropriate preprocessing strategies for achieving optimal results.

4.1. Detailed Analysis of the Best Model:

Seq2Seq LSTM with Attention Mechanism

In this work, the seq2seq LSTM model using attention mechanism is found as the best model for the abstractive summarization in Tigrigna language. Hence the research question of this thesis got positive answer as some deep-learning models are able to summarize given Tigrigna texts to a minimum requirement. The selected model has the ability to handle Tigrigna's complex linguistic structures and limited data organized for abstractive summarization. The Seq2Seq framework allows the model to generate new summaries that conserves the meaning of the input text while transforming it into new abstract with the same meaning. The attention mechanism further enhances this by enabling the model to selectively focus on the most relevant parts of lengthy input sequences, improving the coherence and relevance of generated summaries.

Moreover, the use of fast-Text embedding using down-sampling strengthens the Seq2Seq model by providing sub-word information. Having sub-word information is crucial in morphologically rich languages like Tigrigna. It also helps the model recognize word morphemes and relationships, even the data is limited. This combined organization of the Seq2Seq LSTM with attention and with fast-Text embeddings creates a model that can handle Tigrigna's lengthy input sequences, morphological complex pattern, and data scarcity, ultimately achieving better summarization result and predicting junior abstractive summaries. Though there are numerous challenges, this combination provides a promising foundation for future trends in Tigrigna abstractive summarization.

As already discussed in Table 4 the performance of each model has been compared in their accuracy, loss and ROUGE results. Though it is not excellent result, seq2seq LSTM model with attention and fastText embedding is better over the others.

The overall result of the model's training and validation can also be described using line charts generated using deep-learning packages after the model execution completed. So, shows the line charts for both accuracy and loss with training and validation progresses in 30 epochs respectively.

Figure 3

Text of Specified Style in Document.3-Line Chart

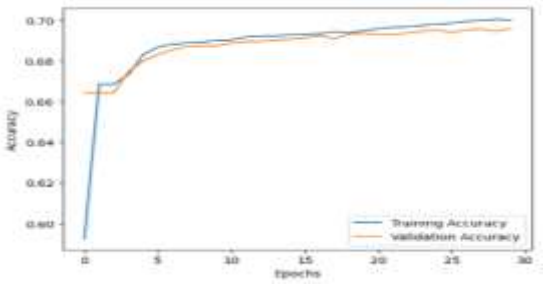
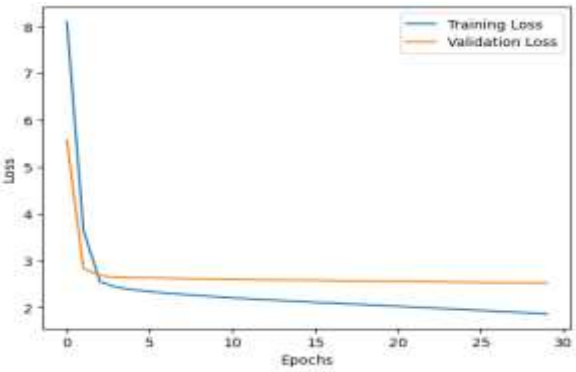


Figure 4

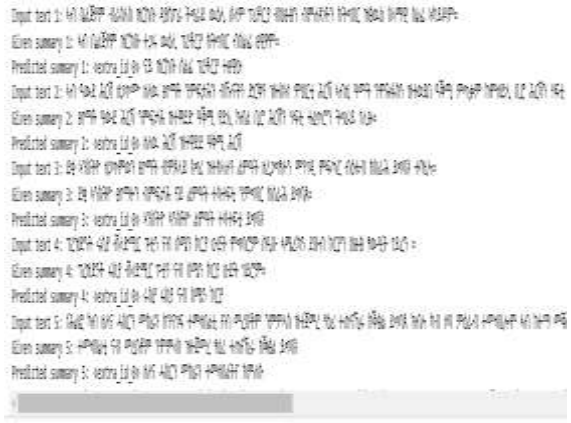
Line Chart of Loss for Seq2seq LSTM Model with Attention



Although an interactive user interface was not developed to present the generated summaries alongside the input texts and reference (human) summaries, sample outputs of the model were documented. While the model shows good results, it also has notable limitations. One common issue is that the model occasionally generates repetitive words, indicating difficulties in maintaining coherence and diversity within the output. It also produces overly short summaries, sometimes consisting of just a single word or, in rare cases, generating no output at all (null summaries). These issues show us the model is still struggling with the task of producing summaries that semantically close the human summaries, highlighting areas where further refinement and tuning may be good to make it better. So, the following sample outputs in

Figure 5

Taken Directly from Output Generated by Seq2seq LSTM Model



4.2. Detailed Analysis of Transformer

All the steps done to analyze the seq2seq LSTM model are also applied to transformer. However, this model has shown less performance than seq2seq LSTM model over all the hyperparameters. **Figure 5** shows sample output of transformer model by taking the raw text and actual summary.

Figure 6

A Sample Output from Transformer Model

In this output, actual summary is reference or human summary prepared manually from the long input text. Whereas predicted summary is generated by the model. Besides, the predicted summaries are showing repetition of words or

phrases. This shows the model is struggling to adapt the language structure and generate summaries. Hence, Seq2seq model using LSTM and attention mechanism is preferred over the transformer model.

5. Conclusion

In today's information-rich world, summarizing large volumes of text is crucial for enabling quick and effective access to relevant content. For many languages, including Tigrigna, there is no easy or structured way to condense vast amounts of online text into reader-friendly summaries. In response to this need, an abstractive summarization approach is proposed to the Tigrigna language using deep learning, a task made challenging by the language's complex morphology, limited digital resources, and the absence of pretrained models or datasets. To tackle this, two core deep learning models are trained: a Seq2Seq model with attention and a Transformer-based model, both of which are well-suited for abstractive summarization. The Seq2Seq model enabled effective sequence learning, capturing word dependencies in a way that aligns with Tigrigna's unique syntactic structures, while the Transformer model leveraged self-attention mechanisms for improved contextual understanding over longer sequences. However, these models required a high degree of adaptation to accommodate Tigrigna's rich morphology and unique language constructs, which differ significantly from more commonly studied languages.

When compared with previous studies, which were primarily extractive in nature and relied on statistical or rule-based approaches, the proposed models demonstrate the potential of semantic-level summarization in capturing contextual meaning and generating fluent text. Although the current results, measured through accuracy, loss, and ROUGE scores, remain modest due to limited data, they represent a significant advancement beyond earlier extractive approaches that could not generalize semantic relationships within Tigrigna sentences. The Seq2Seq model achieved better performance (accuracy ≈ 0.72, loss ≈ 2.17) than the Transformer model, highlighting its suitability for smaller, structured datasets.

This research therefore marks a foundational contribution to the field of Tigrigna text summarization. While the output quality remains at an initial stage, with room for improvement in fluency and coherence, this work serves as a pioneering benchmark and a practical starting point for future research. This study opens the door to further refinements in summarization quality, by establishing a structured approach to abstractive summarization for Tigrigna, potentially benefiting Tigrigna-speaking communities and advancing the broader field of low-resource language processing. The following are some future directions to drive this field forward.

1. Development of a standardized Tigrigna corpus**:** one of the key challenges faced was the lack of a standardized,

annotated corpus for Tigrigna, which limited the model's training potential. Future research should focus on creating a well-structured Tigrigna corpus to support researchers in evaluating and benchmarking their systems more effectively.

2. Exploration of advanced deep learning models: this research employed an encoder-decoder LSTM-based Seq2Seq model for summarization. To push beyond this framework, future studies could explore more complex deep learning architectures, such as Transformer variants, that may capture longer-range dependencies and nuanced contextual details more effectively in Tigrigna.
3. Expanded word embeddings with root words: the word embeddings used in this study cover a limited set of Tigrigna word roots. Building on this work, future research should focus on expanding embeddings to encompass a more comprehensive array of Tigrigna root words, which would enhance the model's language representation and support better generalization.

References

Abdullahi, S. S., Yiming, S., Muhammad, S. H., Mustapha, A., Aminu, A. M., Abdullahi, A., Bello, M., & Aliyu, S. M. (2021). Deep Sequence Models for Text Classification Tasks. *3rd International Conference on Electrical, Communication and Computer Engineering, ICECCE 2021*, *June*, 12–13. https://doi.org/10.1109/ICECCE52056.2021.9514261

Birhanu G, W. M. (2017). College Of Natural Science School of Information Science Automatic Text Summarizer for Tigrinya Language Automatic Text Summarizer for Tigrinya Language. 1–96.Birhanu, G. A. (2017). *Automatic text summarizer for Tigrinya language*. 1–86.

Bo, T., Li, W., & Liu, Y. (2025). A Technical Review of Sequence-to-Sequence Models. *Academic Journal of Natural Science*, *2*(2). https://doi.org/10.70393/616a6e73.323834

Carenini, G., Chi, J., & Cheung, K. (2006). Extractive vs . NLG-based Abstractive Summarization of Evaluative Text : The Effect of Corpus Controversiality. *INLG '08: Proceedings of the Fifth International Natural Language Generation Conference*, http://aclweb.org/anthology-new/W/W08/W08-1106.pdf

Egger, R. (2022). Text Representations and Word Embeddings: Vectorizing Textual Data. *Tourism on the Verge*, *Part F1051*(February), 335–361. https://doi.org/10.1007/978-3-030-88389-8_16

Khalil, M. I. (2020). Abstractive Text Summarization. *Journal of Xidian University*, *14*(6), https://doi.org/10.37896/jxu14.6/094

Paritosh Marathe. (2020). Comprehensive Survey on Abstractive Text Summarization. *International Journal of Engineering Research And*, *V9*(09),

https://doi.org/10.17577/ijertv9is090466

Regassa, M. G., Getachew, M., Advisor, R., & Assabie, Y. (2017). *Topic-based Tigrigna Text Summarization Using WordNet*.

Relan, S., & Rambola, R. (2022). A review on abstractive text summarization Methods. *2022 13th International Conference on Computing Communication and Networking Technologies, ICCCNT*. Doi.org/10.1109/ICCCNT54827.2022.9984332

Sanjabi, N. (2014). *Abstractive Text Summarization with Attention-based Mechanism*.

Sciences, C. (2025). *College of Natural and Computational Sciences School of Information Science*. 6–11.

Shafiq, N., Hamid, I., Asif, M., Nawaz, Q., Aljuaid, H., & Ali, H. (2023). Abstractive text summarization of low- resourced languages using deep learning. *PeerJ Computer Science*, *9*. https://doi.org/10.7717/peerj-cs.1176

Shakil, H., Farooq, A., & Kalita, J. (2024). Abstractive text summarization: State of the art, challenges, and improvements. *Neurocomputing*, *603*. https://doi.org/10.1016/j.neucom.2024.128255

Tamiru, M., & Libsie, M. (2009). *Automatic Amharic Text Summarization Using Latent Semantic Analysis*. *1–106*, Angeles, L., Advocacy, S., Location, O. (2002).

tigrinya · GitHub Topics · GitHub. (n.d.). Retrieved October 15, 2025, from https://github.com/topics/tigrinya

Wazery, Y. M., Saleh, M. E., Alharbi, A., & Ali, A. A. (2022). Abstractive Arabic Text Summarization Based on Deep Learning. *Computational Intelligence and Neuroscience*,*2022*. Doi.org/10.1155/2022/1566890

Yirdaw, E. D. (2011). *Topic-based Amharic Text Summarization*. *June*, Angeles, L., Advocacy, S., Location, O. (2002).

Yirdaw, E. D., & Ejigu, D. (2012). Topic-based amharic text summarization with probabilisic latent semantic analysis. *Proceedings of the International Conference on Management of Emergent Digital EcoSystems, MEDES 2012*, 8–15. https://doi.org/10.1145/2457276.2457279