Journal of Advanced College of Engineering and Management, Vol. 8, 2023

Short Updates- Machine Learning Based News Summarizer

Dr. Raksha Dangol[1], Prashna Adhikari[2], Pranjal Dahal[3], Hrizu Sharma[4]

Assistant Professor, Advanced College of Engineering and Management, Kathmandu [1]

Undergraduate student, Advanced College of Engineering and Management, Kathmandu [2,3,4]

Email:joinwithpraashna@gmail.com [2]

Abstract

Automated Text Summarization is becoming important due to the vast amount of data being generated. Manual processing of documents is tedious, mostly due to the absence of standards. Therefore, there is a need for a mechanism to reduce text size, structure it, and make it readable for users. Natural Language Processing (NLP) is critical for analyzing large amounts of unstructured, text-heavy data. This project aims to address concerns with extractive and abstractive text summarization by introducing a new neural network model that deals with repetitive and incoherent phrases in longer documents. The model incorporates a novel Seq2Seq architecture that enhances the standard attentional model with an intra-attention mechanism. Additionally, a new training method that combines supervised word prediction and reinforcement learning is employed. The model utilizes a hybrid pointer-generator network, which distinguishes it from the standard encoder-decoder model. This approach produces higher quality summaries than existing models.

Keywords: *TF-IDF, Deep learning, Term frequency, Inverse Document Frequency, Automated Text Summarization, Seq2Seq, Pointer-generator, novel intra-attention mechanism*

1. INTRODUCTION

Text summarization is the process of creating a concise and accurate summary of a longer text document. [1] This project aims to develop an application that can identify the most important information from Nepali and English news articles and present it in a human-readable format. To achieve this, we utilize various techniques such as TF-IDF vectorization, Word Embedding, Encoder Decoder model, attention mechanism, and both extractive and abstractive summarization. Extractive summarization is preferred for maintaining factual accuracy and grammatical correctness, while abstractive summarization is better for producing more concise and coherent summaries. [6] By combining these techniques, our model can produce informative and readable summaries. We introduce a key attention mechanism and a new learning objective to address the repeating phrase problem:

i. We use an intra-temporal attention in the encoder that records previous attention weights for each of the input tokens while a sequential intra-attention model in the decoder takes into account which words have already been generated by the decoder.
ii. We propose a new objective function by combining the maximum-likelihood cross-entropy loss used in prior work with rewards from policy gradient reinforcement learning to reduce exposure bias.

It also includes a pointer mechanism to handle Out of Vocabulary (OOV) words, as described in a paper *Get To The Point: Summarization with Pointer-Generator Networks*. The overall goal of the model is to generate high-quality and readable summaries from input texts, even if the input texts are longer and more complex than what previous models could handle.

2. LITERATURE REVIEW

In paper [5] the emphasis of the recent neural system's way to deal with an outline area is characterised as either sentence- extractive, choosing a lot of sentences as the summary, or abstractive, creating the summary from a seq2seq model. In this work, they present a neural model for a single-record summary dependent on joint extraction and pressure. Following later fruitful extractive models, they outline the summarization issues as a progression of local choices. This model picks sentences from the report and after that chooses which of a set of compression options to apply to each chosen sentence. They compute this arrangement of discrete compression rules dependent on the syntactic constituency parser; however, the proposed methodology is measured and utilised for any accessible source of compressions. The limited scope of clustering does not capture the full content and context of the document.

Krishnaveni et.al (2017) [7] proposed a heading-based text summarization method that ranks each sentence based on its relevance to the heading and retrieves the top sentences according to the compression ratio. The model uses unsupervised methods based on semantic similarity measures to score sentences and a set of rules and heuristics to select and order sentences. However, this method cannot capture conflicting information or multiple perspectives present in the text.

Neural encoder-decoder models are widely used in NLP applications such as machine translation , summarization[16], and question answering [17]. These models use recurrent neural networks (RNN), such as long-short term memory network (LSTM) to encode an input sentence into a fixed vector, and create a new output sequence from that vector using another RNN. To apply this sequence-to-sequence approach to natural language, word embeddings[18] are used to convert language tokens to vectors that can be used as inputs for these networks. Attention mechanisms (Bahdanau et al., 2014) make these models more performant and scalable, allowing them to look back at parts of the encoded input sequence while the output is generated. These models often use a fixed input and output vocabulary, which prevents them from learning representations for new words. One way to fix this is to allow the decoder network to point back to some specific words or sub-sequences of the input and copy them onto the output sequence [20].

In paper [19] this pointer mechanism is combined with the original word generation layer in the decoder to allow the model to use either method at each decoding step. The pointer network [20] is a sequence-to- sequence model that uses the soft attention distribution of Bahdanau et al. (2015) to produce an output sequence consisting of elements from the input sequence. The pointer network has been used to create hybrid approaches for NMT [21], language modelling [19], and summarization. Our approach is close to the Forced-Attention Sentence Compression model and the CopyNet model [22] with some small differences.

3. METHODOLOGY

Here are the steps to generate a summary of a document using an encoder-decoder model along with word embedding and TF-IDF:

1. **Data collection**: Scrape the news article from the various news websites then parse it to extract the title, text, link, and published date.
2. **Preprocess the document:** Tokenize the document into a list of individual words and remove any stop words and perform lemmatization.
3. **Load the word embedding:** Use a word embedding model i.e word2vec to create the word embedding for each word in the document to obtain the sentence vectors.
4. **Calculate the TF-IDF and similarity scores:** Perform TF-IDF vectorization to obtain the centroid vectors and calculate the similarity scores for the centroid vectors and sentence vectors.
5. **Extractive Summarization:** Generate extractive summary using the highest similarity score.
6. **Encode the document**: The sentences with highest similarity scores are used as input to the encoder to produce context vector/ latent representation.

7. **Decode the latent representation:** Use the decoder to take the latent representation as input and generate the summary of the document. The decoder network is initialized with the context vector from the encoder network and generates the summary one word at a time. At each time step, the decoder generates a probability distribution over the vocabulary of words and decides whether to output a new word from the vocabulary or copy a word from the input sentence using the attention mechanism and pointer generator.
8. **Evaluate the summary:** Use evaluation metrics- ROUGE to measure the quality of the generated summary and adjust the model's parameters accordingly.
9. **Generate summaries for new documents**: Once the model is trained and evaluated generate summaries of new documents by following the steps 1 to 8.

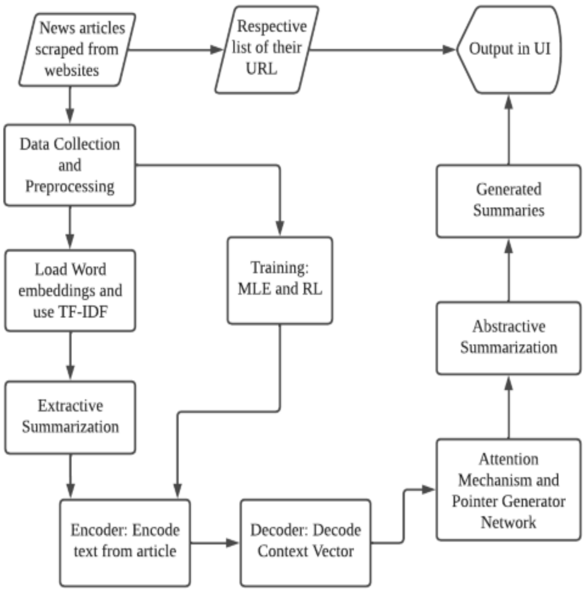

Figure 1: Workflow of News Summarization

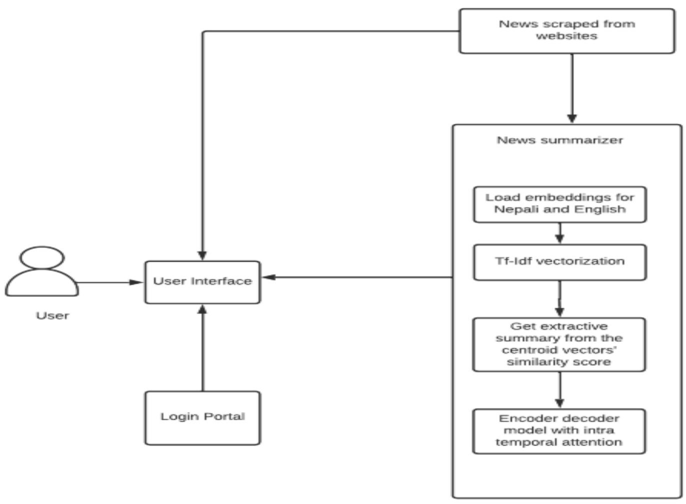

Figure 2 : System Architecture of Short Updates

The main goal is to generate a summary of a given input text by extracting the most important information from the text and condensing it into a shorter form. The summarizer takes x as input and outputs a shortened sentence y.

3.1 Data Collection

3.1.1 News Scraping

News scraping is a subset of web scraping that mainly targets public news websites. It automatically extracts news updates and releases from news articles and websites. We achieve this by scraping news articles from a list of websites specified in a dictionary along with the RSS feed provided, and then use the feed parser library to parse the feed and extract the articles.

For each article, we download the article and parse it to extract the title, text, link, and published date. These details are then stored in a dictionary and added to a list of articles for the specific news website. We only take a number of articles scraped from each website to prevent overloading the server. We also handled exceptions for articles that fail to download or do not have a published date.

3.2 Data Cleaning

3.2.1 Tokenization

Tokenization is essentially done by splitting a phrase, sentence, paragraph, or an entire text document into smaller units, such as individual words or terms. The tokens could be words, numbers or punctuation marks. It takes in a text as input and splits it into individual sentences based on the ‘ |’ (a Devanagari script character) separator. It then removes punctuation from each sentence and returns a list of preprocessed sentences for Nepali text and a ‘.’ separator instead for English text. Smaller units are created by locating word boundaries. These are the ending points of a word and the beginning of the next word. These tokens are considered as a first step for stemming and lemmatization.

3.2.2 Stopwords

Stopwords are the words in any language which do not add much meaning to a sentence. They can safely be ignored without sacrificing the meaning of the sentence. We have removed the stopwords from the nltk library. For some search engines, these are some of the most common, short function words, such as the, is, at, which, and on.

3.2.3 Lemmatization

Lemmatization is a method responsible for grouping different inflected forms of words into the root form, having the same meaning. It is similar to stimming, in turn, it gives the stripped word that has some dictionary meaning. Morphological analysis would require the extraction of the correct lemma of each word.

3.3 Data Pre-processing

3.3.1 TF-IDF

From the preprocessed list of words, the TF-IDF value of each word can then be calculated. The equation of TF-IDF can be seen below.

$$TF = \frac{No.\ of\ repetition\ of\ words\ in\ a\ sent}{No. of\ words\ in\ a\ sentence} \qquad IDF = log(\frac{No\ of\ sentence}{No\ of\ sent\ containing\ that\ word})$$

*TF-IDF=TF*IDF*

The value of TF-IDF ranges from zero to one with ten-digit precision. The TF-IDF score of a word is a measure of its importance in the text. After discovering the features from the whole article by using the statistical approach of frequency-inverse document frequency, we took the sums of the TF-IDF vectors for all words in the sentence to get the centroid vector. The centroid vector was used to get the extractive summary of a news article. It is a vector representation of the text in the news article. It is calculated by averaging the vectors of words in the text that have a high tf-idf score.

3.3.2 Word Embeddings and Word2vec Model

Word embeddings are numeric vectors used to represent words in natural language processing (NLP). They allow similar words to have similar representations, and can be trained using techniques like word2vec and GloVe. A pre-trained Word2Vec model with 300-dimensional vectors for over 0.5 million Nepali words and phrases was implemented using the continuous bag-of-words (CBOW) architecture, which predicts the current word based on its surrounding context words. To train the CBOW model, a large dataset of preprocessed texts and corresponding target words was used, and a supervised learning algorithm such as maximum likelihood estimation (MLE) or minimum risk training (MRT) was employed. The trained model can generate embeddings for new words by predicting the target word based on input context words.

3.3.3 Vocabulary mapping

To preprocess data for an abstractive text summarization model, a vocabulary dictionary is built from the training data. This is done by reading in the data, cleaning and tokenizing the text, counting word frequency with a Counter object, and storing the most common words in a dictionary. The resulting vocabulary is then saved to a file using Python's pickle module.

3.3.4 Cosine Similarity:

Cosine similarity is one of the metrics to measure the text-similarity between two documents irrespective of their size in Natural language Processing.
The mathematical equation of Cosine similarity between two non-zero vectors is:

$$Similarity = \cos(\theta) = \frac{A \cdot B}{\| A \| \| B \|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

Cosine similarity is a mathematical metric that measures the cosine of the angle between two n-dimensional vectors projected in a multi-dimensional space. In text analysis, it is used to calculate the similarity between a centroid vector and the vectors generated from the TF IDF output. The centroid vector is updated to be the average of all vectors that have a similarity above a certain threshold with the initial centroid vector, until convergence. This centroid vector represents the "centre" of the set of vectors and can identify the most important words in the input text.

3.4 Encoder-Decoder Architecture

3.4.1 LSTM

Long Short-Term Memory (LSTM) is a type of recurrent neural network that is capable of learning long-term dependencies in data. The LSTM architecture enhances the basic RNN by adding a “cell state” that enables context to be manipulated not just from step to step, but across the entire sequence of steps. The basic structure of an LSTM unit consists of a cell, an input gate, an output gate, and a

forget gate. The cell remembers values over a long period of time, the input gate controls what information is added to the cell, the output gate controls what information is output from the cell, and the forget gate controls what information is removed from the cell.
The LSTM unit processes the input data and updates the cell state, hidden state, and output at each time step. The hidden state and cell state can be used as input to the LSTM unit at the next time step to process the next input data. This allows the LSTM unit to learn long-term dependencies in the data.[14]

3.4.2 Maximum Likelihood Estimation (MLE)

A machine learning model was trained to generate an output sequence for a given input sequence by minimizing the negative log-likelihood loss, or cross-entropy loss. The model was trained on input-output pairs where the output is the correct output for that input. The goal was to accurately generate the ground truth output for a given input. However, the model suffered from exposure bias, generating low-quality output for new inputs. To address this problem, Self-Critic Policy Gradient (SCPG) training was implemented, which combines MLE training with reinforcement learning to improve the model's performance on new inputs.

3.4.3 Self-Critic Policy Gradient (SCPG) Training

After completing MLE training, the model generates its own output sequence, which is compared to the ground truth output. This process is called self-critiquing. The reward for the sequence is calculated using a scoring metric called ROUGE, and the model's parameters are updated using policy gradient reinforcement learning based on this reward. This SCPG training approach helps the model to learn from its own mistakes and generate high-quality output sequences, which improves its performance on new inputs. SCPG has been shown to improve the quality of generated summaries and translations, reduce repetition, and improve coherence compared to models trained only with MLE.

3.4.4 Encoder-Decoder Model /Sequence-to-Sequence Attentional Model with Pointer Generation Mechanism

The Encoder-Decoder architecture is based on the sequence-to-sequence model, which is trained on a large dataset of input texts and corresponding summaries. During training, the model adjusts its internal parameters to minimize the difference between the generated summary and the reference summary using MLE and Self-critic policy gradient training. The model is then used to generate summaries of new input texts by feeding the text through the encoder and using the decoder to generate the summary. The RNN consists of a set of hidden states that are updated at each time step based on a non-linear activation function. The encoder consists of a bidirectional LSTM, while the decoder consists of a unidirectional LSTM with an attention mechanism over the source-hidden states and a softmax layer over the target vocabulary to generate words. The model introduces a novel intra-attention mechanism that attends separately over the input and generated output, and a new training method that combines supervised word prediction and reinforcement learning (RL). The model is trained on input-summary pairs to minimize the difference between generated and reference summaries. The intra-temporal and intra-decoder attention mechanisms allow the model to focus on important parts of the input and output sequences, generating more accurate and coherent output sequences.

3.4.5 Neural Intra-Attention Model

We present our intra-attention model based on the encoder-decoder network . In all our equations, x = {x1, x2, . . . , xn} represents the sequence of input (article) tokens, y = {y1, y2, . . . , yn0} the sequence of output (summary) tokens, and k denotes the vector concatenation operator.
Our model reads the input sequence with a bi-directional LSTM encoder $\{RNN^{e\_fwd}, RNN^{e\_bwd}\}$

computing hidden states $h_i^e = [h_i^{e\_fwd} || h_i^{e\_bwd}]$ from the embedding vectors of x_i. We use a single LSTM decoder RNN^d, computing hidden states h_t^d from the embedding vectors of y_t. Both input and output embeddings are taken from the same matrix W_{emb}. We initialize the decoder hidden state with $h_0^d = h_n^e$.

3.4.5.1 Intra-Temporal Attention on Input Sequence

In the encoder, the intra-temporal attention mechanism records the attention weights for each input token at each time step t. This means that when the decoder generates the next token, it can use the previous attention weights to focus on different parts of the input sequence. This is done by combining the previous attention weights with the current decoder hidden state and previously generated word to calculate the attention distribution over the input sequence. [1] have shown that such an intra-temporal attention can reduce the amount of repetitions when attending over long documents.
We define e_{ti} as the attention score of the hidden input state h_i^e at decoding time step t:

$$e_{ti} = f(h_t^d, h_i^e)$$

where f can be any function returning a scalar e_{ti} from the h_t^d and h_i^e vectors.
We normalize the attention weights with the following temporal attention function, penalizing input tokens that have obtained high attention scores in past decoding steps. We define new temporal scores e'_{ti}. And finally, we compute the normalized attention scores α_{ti}^e across the inputs and use these weights to obtain the input context vector c_t^e:

$$e'_{ti} = \begin{cases} \exp(e_{ti}) & \text{if } t = 1 \\ \frac{\exp(e_{ti})}{\sum_{j=1}^{t-1} \exp(e_{ji})} & \text{otherwise} \end{cases} \qquad \alpha_{ti}^e = \frac{e'_{ti}}{\sum_{j=1}^{n} e'_{ti}} \qquad c_t^e = \sum_{i=1}^{n} \alpha_{ti}^e h_i^e$$

3.4.5.2 Intra-Decoder Attention

While this intra-temporal attention function ensures that different parts of the encoded input sequence are used, our decoder can still generate repeated phrases based on its own hidden states, especially when generating long sequences. To prevent that, we can incorporate more information about the previously decoded sequence into the decoder. Looking back at previous decoding steps will allow our model to make more structured predictions and avoid repeating the same information, even if that information was generated many steps away.
To achieve this, we introduce an intra-decoder attention mechanism. This mechanism is not present in existing encoder-decoder models for abstractive summarization.
For each decoding step t, our model computes a new decoder context vector c_t^d.

$$e_{tt'}^d = f(h_t^d, h_{t'}^d) \qquad \alpha_{tt'}^d = \frac{\exp(e_{tt'}^d)}{\sum_{j=1}^{t-1} \exp(e_{tj}^d)} \qquad c_t^d = \sum_{j=1}^{t-1} \alpha_{tj}^d h_j^d$$

3.4.6 Pointer Mechanism for handling Out Of Vocabulary (OOV) words

The pointer-generator network is a hybrid model that combines the benefits of the baseline sequence-to-sequence model and a pointer network. It enables the model to copy words via pointing and generate words from a fixed vocabulary. When the model encounters an out-of-vocabulary (OOV) word, it uses attention to locate the corresponding position in the input sequence and then uses a pointer to "copy" the information from that position into the output sequence. The model calculates a generation probability at each time step, based on the context vector, decoder state, and decoder input, to choose between generating a word from the vocabulary or copying a word from the input sequence. The model then calculates the probability distribution over the extended vocabulary, taking into account the generation probability and the attention distribution. Overall, the pointer-generator model allows the model to produce OOV words, which can significantly improve model performance.

3.5 User Interface

Flutter is an open-source framework for building mobile applications that work on both Android and iOS platforms. It uses reusable widgets to create UI elements, including buttons, text fields, images, and animations. To implement a news update UI in a Flutter app, a JSON file with news title, ID, summary, URL link, and publisher was created and added using a custom model API. State management, plugins, REST APIs, and Firebase services were used to fetch and decode data. The home screen displays abstractive and extractive summaries of news, along with a "Read More" section to redirect users to the news website. A drawer was also implemented for additional navigation options, while Firebase was used for sign-in and login services.

4. RESULT AND ANALYSIS

To train our summarization network, we collected 285,843 dataset of news articles through web scraping. We used statistical methods for feature extraction, including term frequency-inverse document frequency and word embedding. For extractive summarization, we retrieved real-time news articles using RSS feeds and calculated similarity scores between the centroid vector and sentence vector to select the most similar sentences for the summary. The number of sentences included in the summary was determined by the desired length passed to the function. For abstractive summarization, we trained an encoder-decoder model with bidirectional LSTM and intra-temporal attention mechanisms using transfer learning on a large dataset of both English and Nepali news articles. We used the extractive summary with the highest similarity score as input text for the model, which generated a coherent summary displayed through our summarization application with real-time updates.

4.1 Result

The baseline Seq2Seq models previously implemented in [7] and [14] have the Rouge scores as given below for English text summarization tasks.

Table 1:Rouge Metrics Comparision

Model	ROUGE-1	ROUGE-2	ROUGE-L
Lead-3 (Nallapati et al., 2017)	39.2	15.7	35.5
Abstractive model (Nallapati et al.,2016)	35.46	13.30	32.65

We report the full-length F-1 score of the ROUGE-1, ROUGE-2 and ROUGE-L metrics. For RL and MLE+RL training, we use the ROUGE-L score as a reinforcement reward.

Table 2: Rouge Metric for Nepali

ROUGE-Metric	r	p	f
ROUGE-1	0.33	0.41	0.35
ROUGE-2	0.17	0.21	0.18
ROUGE-L	0.31	0.40	0.34

The ROUGE-1, ROUGE-2 and ROUGE-L scores for the Summary generation task for Nepali news are 0.35, 0.18 and 0.34 respectively far from 40+ ROUGE-1 and ROUGE-L for the same in the English.

4.2 Analysis

We initially trained an Encoder-Decoder model on a Nepali dataset, but the summaries produced had issues with repetition and lack of context. To improve the quality of the summaries, we implemented a hybrid pointer generator network with intra-temporal attention, which showed significant improvement in the coherence of the summaries, better handling of out-of-vocabulary words, and fewer factual errors. We reported limited-length ROUGE recall scores and found that our mixed-objective model outperformed both the extractive and baseline models.

```
for i in range(0,5):
    print("Review:",seq2text(x_tr[i]))
    print("Original summary:",seq2summary(y_tr[i]))
    print("Predicted summary:",decode_sequence(x_tr[i].reshape(1,max_text_len)))
    print("\n")

Review: कार्यवाहक प्रधानमन्त्री एवं गृहमन्त्री बामदेव गौतमले नन्दप्रसाद अधिकारीले अनसन तोड्न सरकारसँग अर्ब रुपैयाँ मागेको खुलासा गरेका छन् । संसदको सामाजिक न्याय तथा मानवअधि
Original summary: अनसन तोड्न नन्दप्रसादले अर्ब मागेका थिए गृहमन्त्री गौतम
1/1 [==============================] - 1s 1s/step
1/1 [==============================] - 0s 323ms/step
1/1 [==============================] - 0s 34ms/step
1/1 [==============================] - 0s 40ms/step
1/1 [==============================] - 0s 31ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 36ms/step
1/1 [==============================] - 0s 32ms/step
1/1 [==============================] - 0s 43ms/step
1/1 [==============================] - 0s 44ms/step
1/1 [==============================] - 0s 94ms/step
1/1 [==============================] - 0s 72ms/step
Predicted summary:  प्रदेश प्रदेश र र र र मृत्यु मृत्यु मृत्यु मृत्यु
```

Figure 3: Baseline Encoder-Decoder Model Output

```
article: नेकपा ( माओवादी केन्द्र ) का अध्यक्ष पुष्पकमल दाहालले एक्लै लडेर निर्वाचन जित्नुपर्छ भन्ने पक्षमा आफूहरू रहेको बताएका छन् ।
चितवनमा आयोजित गरेको पत्रकार सम्मेलनमा दाहालले चितवन क्षेत्र नं # मा आफूले एक्लै निर्वाचन लडेर जित्ने कोसिसमा रहेको पनि बताए । एक्लै निर्वाचन
 जित्न सक्नुहुन्छ ? भन्ने प्रश्नमा दाहालले भने , ' सम्भावना छ भन्ने विश्वासमा छु ।
orignal: आफूहरू एक्लै लडेर निर्वाचन जित्नुपर्छ भन्ने पक्षमा : अध्यक्ष दाहाल
predicted: एक्लै लडेर निर्वाचन जित्नुपर्छ : दाहाल

article: पर्सामा मोटरसाइकलको ठक्करबाट एक वृद्धको मृत्यु भएको छ । सिमराबाट वीरगन्जतर्फ आउँदै गरेको ना # प #### मोटरसाइकलले
 जितपुरमा गएराति ठक्कर दिँदा गढीमाई नगरपालिका - # का ## वर्षीय वाजुल हकको मृत्यु भएको हो । मोटरसाइकलको ठक्करबाट घाइते भएका
हकको वीरगन्जको नेसनल मेडिकल कलेजमा उपचारका क्रममा मृत्यु भएको जिल्ला प्रहरी कार्यालय पर्सा प्रहरीले जानकारी दिएको छ ।
orignal: पर्सामा मोटरसाइकलको ठक्करबाट एक वृद्धको मृत्यु
predicted: पर्सामा मोटरसाइकलको ठक्करबाट वृद्धको मृत्यु

article: ललितपुर–नेपाल शिक्षक महासंघका अध्यक्ष बाबुराम थापा चारजना विद्यार्थीका अभिभावक बनेका छन् । उनले बुधवार चारजना विद्यार्थीहरुको
अभिभावकत्व ग्रहण गर्दै रुद्रायणी मावि खोकना ललितपुरमा भर्ना गरिदिएका हुन् । नेपाल राष्ट्रिय शिक्षक संगठनका अध्यक्ष समेत रहेका थापाले दर्जी लामालाई कक्षामा ,
 गोइजिना लामालाई केजीमा , रविन कार्कीलाई कक्षा # मा र प्यासिफिक महर्जनलाई कक्षा # मा भर्ना गरिदिएका हुन् ।
orignal: शिक्षक महासंघका अध्यक्ष थापा बने चार विद्यार्थीका अभिभावक
predicted: बाबुराम थापा चारजना विद्यार्थीका अभिभावक

article: संघीय मामिला तथा सामान्य प्रशासनमन्त्री लालबाबु पण्डितले स्वेच्छिक अवकाशका लागि कर्मचारीले दिएको निवेदन स्वीकृत नहुने बताएका छन् ।
 उनले कर्मचारी समायोजन नरोकिने समेत स्पष्ट पारे । ' अघिल्लो सरकारले ल्याएको स्वेच्छिक अवकाश प्याकेजका लागि कर्मचारीहरुले निवेदन दिएका छन् ,
 तर त्यो स्वीकृत हुँदैन ' , स्वेच्छिक अवकाश प्याकेज के हुने भन्नबारे अन्योलता देखिरहेका बेला मन्त्री पण्डितले भने ।
orignal: मन्त्री र सचिव कहाँ धाउँदैमा कर्मचारी समायोजन रोकिँदैनः मन्त्री पण्डित
predicted: स्वेच्छिक अवकाश प्याकेजका लागि निवेदन स्वीकृत हुँदैनः मन्त्री पण्डित
```

Figure 4 : Nepali Output of Seq-to-Seq Attentional Model with Pointer Generation Mechanism

```
orignal: former wisconsin sheriff 's deputy andrew steele , 40 , has been found not legally responsible
in the killing of his wife and sister-in-law . his defense team had argued that lou gehrig 's disease had
damaged his brain , making him not criminally responsible for the deaths .prosecutors believed steele had
planned the killings abnd a third sister said he had displayed inappropriate feelings towards tollefsbol .
predicted: forty-year-old andrew steele pleads not guilty by reason of mental disease .

orignal: study from commerce department says us was worth $ 26trillion in 2006. dc and new jersey worth the
most per acre , with wyoming the least .forty-seven per cent of the us is farmland , only 5.8 per cent developed .
predicted: study : all the land in 48 states was worth a combined $ 23trillion

orignal: head of coffee at pact coffee will corby recommends the best pairings .have a crumbly , hard cheese like
 lancashire with your irish coffee . </t> <t> a delicately spiced pastry will let you detect light espresso flavour
 in latte .
predicted: <t> the new iphone app has been purchased by the wine system .

orignal: <t> the top four tiers of the english game will stand for a minute 's silence to mark the 30th anniversary .
 main stand at bradford city 's valley parade was ablaze in harrowing scenes on may 11 , 1985 . football league
 supporting efforts to raise # 300,000 for the plastic surgery and burns research unit at the university of bradford .
predicted: <t> english football beat mark the 30th anniversary of english football .

orignal: kamron taylor was waiting to be sentenced for the 2013 murder of nelson williams jr , 21 . recaptured on friday
 with loaded gun in chicago after police tip-off .  he fled illinois prison on wednesday after being guard unconscious .
  was seen on cctv wearing guard 's uniform and driving his suv . fears he was hunting down victims family after shouting
threat in court .  he is now in custody on weapons charges .
predicted: kamron taylor was caught in south side , chicago , with a loaded gun
```

Figure 5 : English Output of Seq-to-Seq Attentional Model with Pointer Generation Mechanism

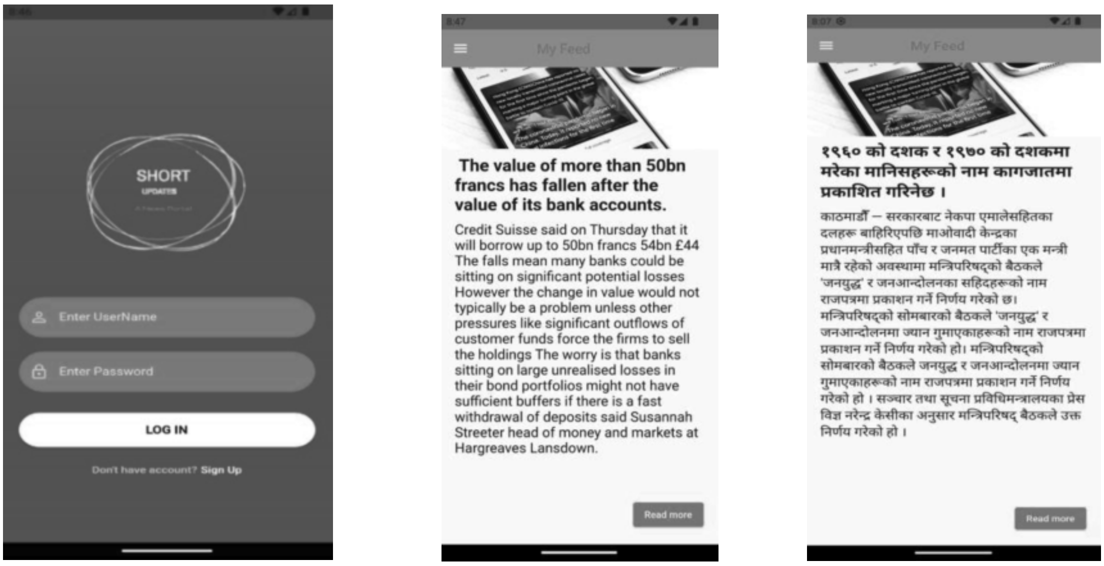

Figure 6: User Interface

5. CONCLUSION

To conclude, the project developed a new automated text summarization model using a hybrid pointer-generator network and an intra-temporal attention mechanism. The model was trained on Nepali and English news articles and evaluated using ROUGE metrics. Results show that the new model outperforms previous abstractive models in generating summaries that are accurate and coherent, although the challenge of achieving higher levels of abstraction remains an open research question. Overall, the project highlights the potential of automated text summarization in processing large amounts of unstructured data and making it more accessible and digestible for users.

6. SUGGESTIONS AND RECOMMENDATIONS

The suggestions for improving the news summarization app include:

1. Offer summaries in multiple languages from diverse sources.
2. Introduce a personalized subscription plan with a recommendation system to suggest relevant news articles to users.
3. Enhance the app to provide topic-specific summaries, allowing users to quickly browse news in their areas of interest.
4. Use machine learning for image and video analysis, enabling the app to extract and summarize relevant information from visual media.
5. Implement text-to-speech functionality to provide automatic reading of summaries, improving accessibility and user experience.

REFERENCES

[1] Digitalcommons.calpoly.edu, 2022. [Online]. Available: https://digitalcommons.calpoly.edu/cgi/viewcontent.cgi?article=1273&context=cpesp

[2] Brownlee, "A Gentle Introduction to Text Summarization", Machine Learning Mastery, 2022. [Online]. Available: https://machinelearningmastery.com/gentle-introduction-text summarization/.

[3] Cse.anits.edu.in, 2022. [Online]. Available:http://cse.anits.edu.in/projects/projects1920B5.pdf.

[4] Nallapati, Ramesh, et al. "Abstractive text summarization using sequence-to-sequence RNNs and beyond." Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning.

[5] K. Karpagam, A. Saradha, A framework for intelligent question answering systems using semantic context-specific document clustering and Wordnet. S ̄adhan ̄a44(3), 62 (2019)

[6] Automatic Text Summarization using Natural Language Processing.pdf - PDFCOFFEE.COM", pdfcoffee.com, 2022

[7] P. Krishnaveni and S. R. Balasundaram, "Automatic text summarization by local scoring and ranking for improving coherence," 2017 International Conference on Computing Methodologies and Communication (ICCMC), Erode, 2017, pp. 59-64. doi: 10.1109/ICCMC.2017.8282539

[8] Stop Words in NLP", Medium, 2022. [Online]. Available: https://medium.com/@saitejaponugoti/stop-words-in-nlp-5b248dadad47.

[9] Intro to Word Embeddings and Vectors for Text Analysis.", Shanelynn.ie, 2022. [Online]. Available: https://www.shanelynn.ie/get-busy-with-word-embeddings introduction/.

[10] B. Kanani, "Cosine Similarity - Text Similarity Metric - Machine Learning Tutorials", Machine Learning Tutorials, 2022.

[11] H. Christian, M. Agus and D. Suhartono, "Single Document Automatic Text Summarization using Term Frequency-Inverse Document Frequency (TF-IDF)", ComTech: Computer, Mathematics and Engineering Applications, vol. 7, no. 4, p. 285, 2016.

[12] Word embedding, [online]. Available:https://en.wikipedia.org/wiki/Word_embedding

[13]LSTM and its Equations, Medium,2022 [online]. Available: https://medium.com/@divyanshu132/lstm-and-its-equations-5ee9246d04af

[14] Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. Proceedings of the 31st AAAI conference, 2017

[15] S. Chopra, M. Auli, and A. M. Rush, "Abstractive sentence summarization with attentive recurrent neural networks," in Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), 2016.

[16] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, "Teaching machines to read and comprehend," in Advances in Neural Information Processing Systems (NIPS), 2015.

[17] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014.

[18] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in Advances in Neural Information Processing Systems (NIPS), 2013

[19] S. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks," in Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS 2014), Montreal, QC, Canada, Dec. 2014, pp. 3104-3112.

[20] C. Gulcehre, S. Ahn, R. Nallapati, B. Zhou, and Y. Bengio, "Pointing the unknown words," in Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL), 2016.