

Evaluating Transformers with NepSum: A Large-Scale Dataset for Nepali Abstractive Summarization

Saskar Khadka and Pawan Khatri

Independent Researchers

Lalitpur, Nepal

{saskar.khadka, pawankhatri8888}@gmail.com

Abstract

Recently, Transformer-based models have significantly improved the performance of abstractive summarization across many high-resource languages. However, for low-resource languages like Nepali, progress in this task has been limited by the lack of annotated data and language-specific pretrained models. To address this gap, we introduce NepSum, the largest Nepali abstractive summarization dataset to date, created by scraping article and headline pairs from multiple online Nepali news portals. We then evaluate the performance of various transformer models on this task. The model selection includes vanilla Transformer, mBART, mT5, mT0, BLOOMZ, Nepali GPT2, and Nepali GPT2 Instruct. We trained the vanilla Transformer from scratch, fully fine-tuned the smaller Nepali GPT2 models and adapted the larger multilingual models using Low-Rank Adaptation (LoRA). We find that mBART outperforms all the other models across both ROUGE and BERTScore metrics.

1 Introduction

Automatic text summarization is an important application of Natural Language Processing (NLP). The ever-growing ocean of digital textual data has made its importance even more evident. Automatic summarization involves generating a concise version of a source document while preserving its key information. It can be categorized into two different approaches: extractive and abstractive. In extractive summarization, the summaries are generated by extracting the most salient segments from the source document and concatenating them without any modification (Shakil et al., 2024), while abstractive summarization focuses on generating human-like summaries by using new words and phrases that may not appear in the source document (Nallapati et al., 2016). This requires the model to have a contextual understanding of the document and then express its core meaning in a concise and

coherent summary. As a result, abstractive summarization is much more complex and challenging than extractive summarization.

The Transformer-based architectures have been very effective in tackling the task of abstractive summarization (Almohaideed and Azmi, 2025). The readily available standardized datasets (Narayan et al., 2018; Hermann et al., 2015) have further contributed to these advancements. However, these advancements have been limited to high-resource languages, and the lack of annotated datasets and pretrained models has become a major hurdle for low-resource languages like Nepali. Even though a lot of multilingual pretrained models have been developed recently, they often do not support the Nepali language. Besides, these models themselves face challenges of morphological richness, syntactic variation, and lack of annotated datasets in multilingual abstractive summarization (Almohaideed and Azmi, 2025), which is the case for the Nepali language. Regardless, multilingual models such as mT5 (Xue et al., 2021) and mBART (Liu et al., 2020) have been shown to be highly effective in abstractive summarization of low-resource languages like Hindi (Kumar, 2024), Urdu (Masih et al., 2025; Tahir et al., 2025), and Bengali (Hayat et al., 2023). For Nepali, existing work on abstractive summarization remains limited. Only a few studies have explored Transformer-based models (Dhakal and Baral, 2025; Bhattarai et al., 2025; Thapa et al., 2025). However, the limited selection of models and the use of non-standardized datasets have left a significant gap in research.

This study aims to address these limitations by establishing a performance benchmark for Nepali abstractive summarization. We achieve this by first overcoming the problem of data scarcity. For this, we compiled a novel abstractive summarization dataset, NepSum, by scraping news articles and titles from multiple online news portals. The dataset consists of approximately 368K unique

article-headline pairs, making it comparable to the widely used English benchmark datasets, such as CNN/DailyMail (Hermann et al., 2015), and XSum (Narayan et al., 2018) in terms of scale. We then use NepSum to evaluate a diverse set of Transformer-based models to establish a baseline for future research. We first trained the vanilla Transformer (Vaswani et al., 2017) from scratch, and then fine-tuned a range of pretrained models. For efficient fine-tuning, the larger multilingual models, mBART (Liu et al., 2020), mT5 (Xue et al., 2021), and the instruction-tuned BLOOMZ and mT0 (Muenighoff et al., 2023) were adapted using LoRA (Hu et al., 2021), whereas the relatively smaller Nepali GPT2 (Thapa et al., 2025) models were fully fine-tuned. These models allowed us to compare performance across different architectures, model sizes, and pretraining objectives, which we report as ROUGE (Lin, 2004) score and BERTScore (Zhang et al., 2020).

2 Related Works

Early research on automatic text summarization of Nepali language articles focused on extractive approaches. The introduction of Sequence-to-Sequence models (Sutskever et al., 2014) and attention mechanism (Bahdanau et al., 2016) enabled the shift towards abstractive approaches. Recently, the Transformer-based (Vaswani et al., 2017) models have replaced the RNNs as the primary architecture across a wide variety of NLP tasks including summarization.

Mishra et al. (2020) were the first to investigate Nepali abstractive summarization. They utilized an encoder-decoder architecture and trained the model to generate Nepali news headlines using a dataset of approximately 138K article-headline pairs. To overcome the vanishing and exploding gradient problem in traditional RNNs, they used a Bidirectional GRU in the encoder and a unidirectional GRU in the decoder. The word embeddings were trained using the FastText continuous bag-of-words model. To decrease the number of parameters, they formulated the task as regression rather than classification, so each decoder step would generate a word embedding instead of the probability distribution across the vocabulary. Finally, the generated word vectors were converted back to words using FastText to obtain the output headline, which was evaluated using BLEU (Papineni et al., 2002) score. However, this regression-based approach comes with limita-

tions, as slight variations in vectors could result in a completely different word selection.

Timalsina et al. (2022) presents an LSTM-based encoder-decoder architecture with the attention mechanism (Bahdanau et al., 2016). First, they compiled a dataset consisting of approximately 128K article-headline pairs by scraping different Nepali news portals. Using this dataset, they performed a comprehensive empirical evaluation of nine different models with varying encoder layers and RNN hidden units. Among them, a single layered encoder-decoder model with 256 hidden units achieved the highest ROUGE (Lin, 2004) score.

Earlier studies relied on RNN-based architectures that are limited by their ability to capture long-term dependencies, which is crucial for abstractive summarization. While the inclusion of attention mechanism helped mitigate this problem, the sequential nature of RNN still prevented parallelization, making them computationally very expensive to train as context size grows. To overcome these challenges, modern transformer architectures rely on multi-head self-attention (Vaswani et al., 2017). Dhakal and Baral (2025) introduced these modern transformer architectures to Nepali abstractive summarization. Specifically, they utilized two multilingual pretrained transformers, mT5 (Xue et al., 2021) and mBART (Liu et al., 2020). They fine-tuned these models using Parameter-Efficient Fine-Tuning (PEFT) techniques such as LoRA (Hu et al., 2021) and QLoRA (Dettmers et al., 2023) on a dataset of 72K news article-headline pairs. They evaluated their models using the ROUGE (Lin, 2004) metric as well as human evaluators, and report the 4-bit quantized mBART with LoRA as the best performing model.

Thapa et al. (2025) utilized a monolingual GPT model for abstractive summarization. They trained the GPT-2 architecture (Radford et al., 2019) on a corpus of 30.4 GB Nepali textual data. They further instruction tuned this model, and fine-tuned both the pretrained and instruction-tuned variants on the Nepali subset of the XL-Sum dataset (Hasan et al., 2021), consisting of around 7.26K samples. The models were evaluated using the ROUGE (Lin, 2004) metric. The authors report that the models tend to hallucinate for longer sequences resulting in relatively lower ROUGE scores.

3 Methodology

3.1 Data Collection

For this study, we compiled NepSum, a large-scale Nepali summarization dataset consisting of approximately 368K samples. To ensure diversity in content and writing styles, we scraped news articles and headlines from 12 popular online Nepali news portals across 10 different categories.

3.2 Data Cleaning

After compiling the dataset, we processed it through a data cleaning pipeline, where we removed null entries, duplicates, HTML tags, emails, emojis, and English characters (a-z, A-Z). We also discarded very short articles, and to maintain consistency with the Devanagari script, we replaced the Arabic numerals (0–9) with Devanagari numerals (०–९).

Depending on the model, we handled the special characters differently. For fine-tuning, they were kept intact, considering the ability of pretrained transformers to handle them effectively. However, to train the Transformer from scratch, we removed all special characters except %, ., ?, \$, and - as they carry task-relevant significance within the articles and headlines (e.g., in Nepali words दुःख (grief) and पुनः (again), questions, sports scores, financial figures, etc.).

Finally, for fine-tuning instruction-tuned models, mT0, BLOOMZ, and Nepali GPT2 Instruct, we appended natural-language instruction prompts at the beginning of the articles.

3.3 Dataset Analysis

After processing the dataset, we were left with 348,409 clean article-headline pairs. The dataset was split into training (85%), validation (10%), and test (5%) sets. Table 1 presents the summary statistics of the final dataset. The distribution of article length is positively skewed, with a mean and median length of 236 and 176 words respectively.

3.4 Tokenization

For the vanilla Transformer, we trained a SentencePiece (Kudo and Richardson, 2018) tokenizer using the Byte-Pair Encoding (BPE) algorithm (Sennrich et al., 2016) on the news articles and headlines from the training data, with a vocabulary size of 30,000. The SentencePiece tokenizer creates a vocabulary of subwords, allowing it to represent unseen words as a combination of known subwords. This makes tokenization much more efficient, especially for

Statistic	Value
Total Samples	348,409
Average words in article	236
Average words in headline	8
Median words in article	176
Median words in headline	7
Training Samples	296,147
Validation Samples	34,493
Test Samples	17,769

Table 1: Summary statistics of cleaned NepSum dataset.

morphologically rich language like Nepali as compared to word-level tokenizers which often result in large vocabularies and many out-of-vocabulary (OOV) tokens. This tokenizer was used to tokenize the input articles and target summaries to a maximum length of 512 and 16 tokens, respectively. We used a shorter 16-token limit for summaries, as we needed to train our decoder from scratch, and shorter sequences would simplify training and result in faster convergence.

For pretrained models, we used their respective pretrained tokenizers. For Nepali GPT2 specifically, the authors noted the tendency of the model to hallucinate with inputs exceeding 400 tokens (Thapa et al., 2025). Therefore, we limited the maximum combined article-headline length to 400 tokens, using 374 tokens for articles and 26 tokens for the headline. For the rest of the models, we limited the maximum lengths to 512 tokens for articles and 26 tokens for summaries. Table 2 presents the average sequence lengths and tokens-to-words ratios across different model tokenizers.

3.5 Models

3.5.1 Vanilla Transformer

We implemented the original Transformer architecture as described by Vaswani et al. (2017), with three key modifications. First, we adjusted the model parameters to suit our dataset and computational resources. Second, we replaced the ReLU activation function with the GELU (Hendrycks and Gimpel, 2023), which has been shown to improve training stability and model performance (Lee, 2023). Third, we used learned positional encodings instead of sinusoidal ones, allowing the model to generate positional information specific to our dataset. The resulting model has approximately 15 million parameters. The overall model architecture is summarized in Table 3.

Tokenizer	Average Tokens		Average Tokens-to-Words Ratio	
	Article	Headline	Article	Headline
Vanilla Transformer	123	8	1.1	1.1
mBART	396	16	1.68	2.18
BLOOMZ	362	13	1.53	1.78
mT5 / mT0	604	22	2.56	2.89
Nepali GPT2 / Nepali GPT2 Inst.	275	9	1.16	1.23

Table 2: Tokenization Statistics Across Model Tokenizers

Component	Value
Encoder Layers (N_{enc})	4
Decoder Layers (N_{dec})	4
Model Dimension (d_{model})	256
Attention Heads (h)	8
Feedforward Dimension (d_{ff})	1,024
Activation Function	GELU
Encoder Sequence Length (L_{enc})	512
Decoder Sequence Length (L_{dec})	16

Table 3: Vanilla Transformer Architecture Details

3.5.2 Pretrained Models

Abstractive summarization requires a model to generate text sequences, which is possible only through a decoder. We have two choices, an encoder-decoder transformer or a decoder-only transformer. There have not been enough studies on Nepali generative pretrained models. Much of the research has been focused on the non-generative encoder-only architectures. To our knowledge, only two prior studies have been conducted on monolingual Nepali generative models, both of which experimented with the GPT2 architecture (Thapa et al., 2025; Pudasaini et al., 2025). For our work, we used the Nepali GPT2 (Thapa et al., 2025) model as it was trained on a much larger corpus of 30.4 GB and also provides an instruction-tuned model variant, which we also utilized. The lack of monolingual model choices led us to also rely on multilingual models. These models were trained on multiple languages, including Nepali, and have shown promising results in generative tasks. We experimented with four such multilingual models, mBART-large-50, mT5-base, mT0-base and BLOOMZ-560M. While larger, more powerful versions of these models exist, they were not feasible for us to fine-tune using our computational resources. These models allowed us to conduct a thorough model assessment across multiple architectures and pretraining objectives, providing a baseline for future studies in Nepali

generative Transformers and summarization.

- **Nepali GPT2** (Thapa et al., 2025) is a monolingual decoder-only model with 124M parameters. It was pretrained using the Causal Language Modeling (CLM). It has an instruction-tuned variant, **Nepali GPT2 Instruct** created by instruction-tuning the base model on the Nepali alpaca dataset (Saugatkafley, 2024).
- **BLOOMZ-560M** (Muennighoff et al., 2023) is a multilingual decoder-only model with 560M parameters. It is an instruction-tuned version of BLOOM (Workshop et al., 2023), which was pretrained with an objective of Causal Language Modeling (CLM).
- **mBART-large-50** (Liu et al., 2020) is a multilingual encoder-decoder model with 611M parameters. It was pretrained with an objective of denoising the entire input sequence, which was corrupted using word span masking and sentence permutation.
- **mT5-base** (Xue et al., 2021) is a multilingual encoder-decoder model with 580M parameters. It was pretrained using the span corruption objective, where, unlike mBART, the decoder learns to generate only the corrupted spans. The model was instruction-tuned on the xP3 (Muennighoff et al., 2023) dataset to create **mT0-base** (Muennighoff et al., 2023).

3.6 Training and Fine-tuning

3.6.1 Vanilla Transformer

The vanilla Transformer was trained with an objective of summarization. The encoder learned to produce a contextual representation of input text articles, while the decoder was trained to generate summaries autoregressively using teacher forcing. We used the Adam optimizer (Kingma and Ba, 2015) with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-7}$. The learning rate was varied according to the schedule

proposed by Vaswani et al. (2017). We set warmup steps to 5000 and used a batch size of 32. The model was trained for 20 epochs ($\approx 185\text{K}$ steps), at which point the improvement in validation loss plateaued, indicating convergence. Additionally, to prevent overfitting and improve generalization, we employed label smoothing of $\epsilon_{ls} = 0.1$ (Szegedy et al., 2016), dropout of $P_{drop} = 0.2$ (Srivastava et al., 2014) and a weight decay factor of $\lambda = 0.01$.

3.6.2 Pretrained Models

The pretrained models were fine-tuned for 4 epochs using the Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$). We used a linear schedule that increased the learning rate from 0 to 2×10^{-5} during warmup and then decreased linearly to 0 for the remainder of training. To maximize the hardware utilization, we set the maximum possible batch size that would fit within our GPU memory for each model.

For the larger multilingual models, we employed Low-Rank Adaptation (LoRA) (Hu et al., 2021) to make training faster and more resource-efficient. We applied LoRA to the attention projection layers with a rank $r = 16$, and a scaling factor $\alpha = 32$. In contrast, the smaller Nepali GPT2 models were fully fine-tuned. For regularization, a weight decay of $\lambda = 0.01$, and a dropout of 0.1 was also applied.

3.7 Evaluation Metrics

The ROUGE (Lin, 2004) metric scores the quality of summarization by measuring the lexical overlap between the generated and reference summaries. Since it relies on exact n-gram match, it fails to account for synonyms and morphological variations, and penalizes semantically correct outputs that use different phrasing. Despite these limitations, it is still used as a standard metric for evaluating summarization due to its established baselines, interpretability and computational efficiency (Almohaimeed and Azmi, 2025). So, alongside ROUGE, we also decided to use the BERTScore (Zhang et al., 2020) metric which uses the mBERT (Devlin et al., 2019) embeddings to measure the semantic similarity between the outputs and the reference human summaries. Using these two metrics, we can quantitatively measure both the text overlap accuracy as well as semantic similarity, providing a comprehensive evaluation of model performance.

3.8 Experimental Setup

Data collection was performed on a local machine using the Selenium API for Python. The training and fine-tuning procedure required significant computational resources and was not feasible with our local hardware capabilities. For this, we utilized Kaggle’s freely available cloud computing environment. The environment came with 30 GB of RAM and a P100 GPU with 16 GB VRAM. The GPU usage was limited to 30 hours/week, while each computing session was limited to 12 hours. To overcome these constraints, we used model checkpointing to train the model across multiple sessions. Mixed precision FP16 was used for fine-tuning the pretrained models, except for the mT5 and mT0 models as they would result in 0 training loss and NaN validation loss. While switching to BF16 would solve the problem, the available GPU didn’t support it, so they were trained using FP32 precision. The experiments were performed using different Python libraries such as Pandas (McKinney, 2010) for data processing, Tensorflow (Abadi et al., 2016) for setting up and training the vanilla Transformer, and the Hugging Face Transformers (Wolf et al., 2020), and PEFT for fine-tuning the pretrained models.

4 Results

Table 4 summarizes the ROUGE and BERTScore F1 scores for all the tested configurations. The mBART + LoRA model with beam width 5 achieved the highest score across all metrics, indicating its ability to generate summaries that are both lexically and semantically aligned with the reference summaries. The performance was followed closely by the vanilla Transformer and BLOOMZ. Despite having significantly fewer parameters, the Vanilla Transformer at beam width 3 outperformed all the other models in ROUGE scores except for mBART in ROUGE-2, with a marginal difference of 0.05. It also maintained a relatively high BERTScore across different beam configurations. These results suggest that a comparatively small model trained from scratch on a specific task can be as effective as fine-tuning larger pretrained models. However, this does require a sufficiently large training dataset, which is often the major hurdle for low-resource languages.

The mT5 and mT0 models achieved relatively lower scores. While the instruction-tuned mT0 showed a slight improvement over mT5, the overall

Model	ROUGE-1		ROUGE-2		ROUGE-L		BERTScore	
	BW=3	BW=5	BW=3	BW=5	BW=3	BW=5	BW=3	BW=5
Vanilla Transformer	41.10	41.19	22.54	22.61	39.66	39.76	80.96	81.00
Nepali GPT2	37.99	38.57	20.19	20.69	36.57	37.17	79.75	80.02
Nepali GPT2 Inst.	37.58	38.38	19.93	20.73	36.14	36.97	79.83	80.09
mBART + LoRA	40.69	41.19	22.59	23.20	39.32	39.77	81.23	81.33
mT5 + LoRA	36.98	37.04	20.08	20.21	35.63	35.72	79.72	79.73
mT0 + LoRA	38.01	38.07	20.76	20.86	36.62	36.67	80.15	80.17
BLOOMZ + LoRA	39.88	40.41	22.15	22.57	38.42	38.95	81.10	81.21

Table 4: Model Evaluation Results for Beam Width (BW) 3 and 5

performance remained well below the top models. This gap can be attributed to their pretraining objective. As mentioned by Chirkova et al. (2024), mT5 often lags behind mBART in summarization as it is trained to generate short, fragmented masked spans rather than full sequences. In contrast, the BLOOMZ + LoRA model, which was instruction-tuned on the same dataset as mT0, showed competitive performance with higher ROUGE scores, despite it being susceptible to the attention degeneration problem (Fu et al., 2023). These results further consolidate the limitations of the span corruption as a pretraining objective.

The monolingual Nepali GPT2 models also underperformed despite being fully fine-tuned, likely due to their relatively small size and the additional constraints faced by decoder-only models (Fu et al., 2023). While there was a noticeable performance boost in mT0 over mT5, such a boost was not observed in the Nepali GPT2 models. The performance of Nepali GPT2 model was comparable to the performance of its instruction-tuned variant, with one outperforming the other in each metric, with a marginal difference. These results suggest that the impact of instruction tuning was almost negligible in Nepali GPT2 for the task of summarization.

While we achieved varying ROUGE scores, the BERTScore remained relatively high and mostly stable across all models. Manual inspection of a small subset of the generated summaries from the test set also showed that the models generally produced clear and concise summaries with a few instances of text degeneration, hallucination, and truncation. Table 5 and Table 6 present instances of successful and erroneous generations respectively.

- **Text Degeneration:** This issue was specific to the Nepali GPT2 models, where they generated random characters and words at the end of

the sequence. Although there were instances where using a beam width of 5 solved it, the issue occasionally persisted across both configurations. This random text often started with तस्वीर (Picture). News articles use phrases such as १५ तस्वीर (15 pictures), तस्वीरहरु सहित (with pictures) at the end of headlines to indicate that the article contains additional pictures. However, such samples were very rare in our dataset (276 samples). Since the base model was also pretrained using news articles, the model likely learned this pattern during pretraining.

- **Hallucination:** A common problem that haunts almost all generative models. We observed a few instances where the models produced a fluent summary with factually incorrect details.
- **Truncation:** This issue was specifically observed in the mT5 and mT0 models where their outputs were prematurely cut off before completing the summary. As presented in Table 2, their tokenizer resulted in a relatively higher tokens-to-words ratio as compared to others which led to longer target sequences, increasing the likelihood of exceeding the maximum generation threshold. This particular issue was not very significant as simply increasing the threshold would resolve it.

5 Conclusion

In this paper, we introduced NepSum, a large-scale dataset for Nepali abstractive summarization to address the lack of standardized resources for this task. The dataset consists of diverse articles compiled from multiple Nepali news portals across a variety of news categories. Furthermore, to provide a comprehensive performance benchmark, we

	Sample 1	Sample 2	Sample 3	Sample 4
Reference	डल्लु आवास क्षेत्रमा बहुउद्देश्यीय कभर्ड हल बन्ने	प्रदेश नं १ मा ४७९ जनामा कोरोना सङ्क्रमण	२१ वटा दुईपाङ्ग्रे सवारी साधन सम्बन्धित धनीलाई हस्तान्तरण	एसियाडका लागि बक्सिङको तयारी भारतमा
Translation	Multi-purpose covered hall to be constructed in Dallu housing area	479 people infected with COVID-19 in Province 1	21 two-wheeler vehicles handed over to their respective owners	Boxing preparations for Asian Games in India
Vanilla Transformer	डल्लु आवास क्षेत्रमा बहुउद्देश्यीय कभर्ड हल बन्ने	प्रदेश नं १ मा ४७९ जनामा कोरोना सङ्क्रमण	हराएका २१ वटा सवारी साधन खोजी गरी धनीलाई हस्तान्तरण	बक्सिङ टोली भारतमा
mBART	भूकम्प स्मृति बहुउद्देश्यीय कभर्ड हल बन्ने	प्रदेश नं १ मा ४७९ जनामा कोरोना पुष्टि	हराएका २१ वटा दुई पाङ्ग्रे सवारी साधन ट्राफिक प्रहरीलाई हस्तान्तरण	एसियाड तयारीका लागि नेपाली बक्सिङ टोली भारतमा
mT5	डल्लु आवास क्षेत्रमा बहुउद्देश्यीय कभर्डहल बन्ने	प्रदेश १ मा ४७९ जनामा कोरोना सङ्क्रमण पुष्टि	हराएका २१ वटा दुई पाङ्ग्रे सवारी साधन खोजी गरी ट्राफिक प्रहरी कार्य	१९औँ एसियाली खेलकुदका लागि नेपाली बक्सिङ खेलाडीको तयारी भारतमा
mT0	डल्लु आवास क्षेत्रमा भूकम्प स्मृति बहुउद्देश्यीय कभर्डहल बन्ने	प्रदेश १ मा ४७९ जनामा कोरोना पुष्टि	हराएका २१ वटा दुई पाङ्ग्रे सवारी साधन खोजी गरी धनीलाई हस्तान्तरण	१९औँ एसियाली खेलकुदका लागि नेपाली बक्सिङ खेलाडीको तयारी भारतमा
BLOOMZ	डल्लु आवास क्षेत्रमा बहुउद्देश्यीय कभर्ड हल बन्ने	प्रदेश नं १ मा ४७९ जनामा कोरोना सङ्क्रमण	उपत्यकामा हराएका सवारी साधन खोजी गरी धनीलाई हस्तान्तरण	भारतमा हुने एसियाली खेलकुदका लागि बक्सिङ खेलाडीको तयारी
Nepali GPT2	डल्लु आवास क्षेत्रमा बहुउद्देश्यीय कभर्ड हल बन्ने शिलान्यास	प्रदेश नं १ मा ४७९ जनामा कोरोना सङ्क्रमण	हराएका २१ वटा दुई पाङ्ग्रे सवारी साधन धनीलाई हस्तान्तरण	नेपाली बक्सर टोली भारत जाँदै
Nepali GPT2 Inst.	डल्लु आवास क्षेत्रमा बहुउद्देश्यीय कभर्ड हल बन्ने	प्रदेश नं १ मा ४७९ जनामा कोरोना सङ्क्रमण	हराएका २१ वटा सवारी साधन धनीलाई हस्तान्तरण	एसियाली खेलकुदका लागि बक्सिङ खेलाडीको तयारी भारतमा

Table 5: Cases of Successful Generation Across Models with Beam Width 5.

utilized this dataset to evaluate seven different transformer models in terms of textual overlap and semantic similarity. Our results demonstrate the impact of model architecture, model size, and pretraining objectives on summarization quality. We found that the encoder-decoder model with denoising pre-training objective (mBART) consistently outper-

formed all other model configurations. Notably, the vanilla Transformer model that was trained from scratch performed remarkably well despite its small size. Finally, we also observed three distinct failure modes: text degeneration, hallucination, and truncation.

Model	Reference	Ref. English	Generated	Gen. English	Remarks
Vanilla Transformer	वडा अध्यक्ष नेम्बाङले एकै व्यक्तिलाई दोहोरो नागरिकता दिन सिफारिस गरेको खुलासा	Ward Chairperson Nembang found recommending duplicate citizenship for same individual	लमजुङमा एकै दिन दोहोरो नागरिकता सिफारिस	Duplicate citizenship recommended in Lamjung on the same day	Hallucination: Wrong location and timeframe. Incident occurred in Taplejung in 2016 and 2022.
BLOOMZ	अमेरिकामा एक व्याक्तिद्वारा परिवारका सात जनाको हत्यापछि आत्महत्या	American man commits suicide after killing seven family members	पाँच बच्चासहित सात जनाको हत्यापछि श्रीमतीले गरिन् आत्महत्या	Wife commits suicide after killing seven including five children	Hallucination: Wrong perpetrator.
Nepali GPT	बस दुर्घटनामा ३४ जना घाइते	34 injured in bus accident	तनहुँमा बस दुर्घटना हुँदा ३४ जना घाइते तस्वीर) तस्वीर)	34 injured in bus accident in Tanahun Image) Image)	Degeneration: Model generates irrelevant tokens.
mT5	पन्जाबले दिएको ठूलो लक्ष्य पछ्याउँदै राजस्थानको रोमाञ्चक जित	Rajasthan chase down massive target to secure an exciting win over Punjab	पन्जाबले दिएको २ सय २४ रनको ठूलो लक्ष्य भेट्याउँदै राजस्थान रोय	Punjab's 2 hundred 42 run target chased by Rajasthan Roy	Truncation: Model fails to complete Rajasthan Royals.

Table 6: Cases of Text Degeneration, Hallucination and Truncation Across Models with Beam Width 5.

Limitations

Despite our findings, the study has a few limitations. First, we mostly rely on multilingual models due to the limited availability of Nepali pretrained generative models. While these models have demonstrated strong performance, Nepali represents only a small fraction of their pretraining data which may limit their overall ability for low-resource language processing. Second, due to computational constraints, our evaluation was restricted to smaller models and utilized LoRA for fine-tuning the multilingual models to improve computational efficiency. Finally, although the dataset consists of a variety of news sources and categories, it remains limited to the news domain. As a result, the dataset and the performance of the evaluated models may not generalize to other domains, such as legal, medical, or conversational texts. To address these gaps, future work should focus on evaluating larger language models, building monolingual Nepali generative transform-

ers, expanding the dataset beyond the news domain, and providing a more comprehensive analysis of failure modes.

References

- Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, and 3 others. 2016. [Tensorflow: a system for large-scale machine learning](#). In *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation, OSDI'16*, page 265–283, USA. USENIX Association.
- Norah Almohaimeed and Aqil M. Azmi. 2025. [Abstractive text summarization: A comprehensive survey of techniques, systems, and challenges](#). *Computer Science Review*, 57:100762.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Ben-

- gio. 2016. [Neural machine translation by jointly learning to align and translate](#). *Preprint*, arXiv:1409.0473.
- Bidhya Bhattarai, Biplav Chaudhary, Punam Bashyal, Sajita Khadka, and Krishnanand Badu. 2025. [Multilingual transformer-based summarization for low-resource nepali news articles](#). *Journal of Himalaya College of Engineering*, 2(1):76–81.
- Nadezhda Chirkova, Sheng Liang, and Vassilina Nikoulina. 2024. [Empirical study of pretrained multilingual language models for zero-shot cross-lingual knowledge transfer in generation](#). *Preprint*, arXiv:2310.09917.
- Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. [Qlora: Efficient finetuning of quantized llms](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 10088–10115. Curran Associates, Inc.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Prakash Dhakal and Daya Sagar Baral. 2025. [Abstractive summarization of low resourced Nepali language using multilingual transformers](#). In *Proceedings of the First Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2025)*, pages 124–133, Abu Dhabi, UAE. International Committee on Computational Linguistics.
- Zihao Fu, Wai Lam, Qian Yu, Anthony Man-Cho So, Shengding Hu, Zhiyuan Liu, and Nigel Collier. 2023. [Decoder-only or encoder-decoder? interpreting language model as a regularized encoder-decoder](#). *Preprint*, arXiv:2304.04052.
- Tahmid Hasan, Abhik Bhattacharjee, Md. Saiful Islam, Kazi Mubasshir, Yuan-Fang Li, Yong-Bin Kang, M. Sohel Rahman, and Rifat Shahriyar. 2021. [XLsum: Large-scale multilingual abstractive summarization for 44 languages](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4693–4703, Online. Association for Computational Linguistics.
- S. M. Afif Ibne Hayat, Avishek Das, and Mohammed Moshikul Hoque. 2023. [Abstractive bengali text summarization using transformer-based learning](#). In *2023 6th International Conference on Electrical Information and Communication Technology (EICT)*, pages 1–6.
- Dan Hendrycks and Kevin Gimpel. 2023. [Gaussian error linear units \(gelus\)](#). *Preprint*, arXiv:1606.08415.
- Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. [Teaching machines to read and comprehend](#). In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. [Lora: Low-rank adaptation of large language models](#). *Preprint*, arXiv:2106.09685.
- Diederik P. Kingma and Jimmy Ba. 2015. [Adam: A method for stochastic optimization](#). In *International Conference on Learning Representations (ICLR)*.
- Taku Kudo and John Richardson. 2018. [SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.
- Jeetendra Kumar. 2024. [Hindi abstractive text summarization using transliteration with pre-trained model](#). *Journal of Electrical Systems*, 20:2089–2110.
- Minhyeok Lee. 2023. [Gelu activation function in deep learning: A comprehensive mathematical analysis and performance](#). *Preprint*, arXiv:2305.12073.
- Chin-Yew Lin. 2004. [ROUGE: A package for automatic evaluation of summaries](#). In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.
- Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. [Multilingual denoising pre-training for neural machine translation](#). *Transactions of the Association for Computational Linguistics*, 8:726–742.
- Salman Masih, Mehdi Hassan, Labiba Gillani Fahad, and Bilal Hassan. 2025. [Transformer-based abstractive summarization of legal texts in low-resource languages](#). *Electronics*, 14(12).
- Wes McKinney. 2010. [Data structures for statistical computing in python](#). *SciPy 2010*.
- Kaushal Raj Mishra, Jayshree Rathi, and Janardan Banjara. 2020. [Encoder decoder based nepali news headline generation](#). *International Journal of Computer Applications*, 175(20):1–4.
- Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. [Crosslingual generalization through multitask finetuning](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15991–16111, Toronto, Canada. Association for Computational Linguistics.

- Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, Çağlar Gulçehre, and Bing Xiang. 2016. [Abstractive text summarization using sequence-to-sequence RNNs and beyond](#). In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.
- Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. [Don’t give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. [Bleu: a method for automatic evaluation of machine translation](#). In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- Shushanta Pudasaini, Aman Shakya, Siddhartha Shrestha, Sahil Bhatta, Sunil Thapa, and Sushmita Palikhe. 2025. [Nepaligpt: A generative language model for the nepali language](#). *Preprint*, arXiv:2506.16399.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. [Language models are unsupervised multitask learners](#). *OpenAI Blog*.
- Saugatkafley. 2024. alpaca-nepali-sft. <https://huggingface.co/datasets/Saugatkafley/alpaca-nepali-sft>.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. [Neural machine translation of rare words with subword units](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.
- Hassan Shakil, Ahmad Farooq, and Jugal Kalita. 2024. [Abstractive text summarization: State of the art, challenges, and improvements](#). *Neurocomputing*, 603:128255.
- Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. [Dropout: A simple way to prevent neural networks from overfitting](#). *Journal of Machine Learning Research*, 15(56):1929–1958.
- Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. [Sequence to sequence learning with neural networks](#). In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.
- Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. [Rethinking the inception architecture for computer vision](#). In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826.
- Munief Hassan Tahir, Sana Shams, Layba Fiaz, Farah Adeeba, and Sarmad Hussain. 2025. [Benchmarking the performance of pre-trained LLMs across Urdu NLP tasks](#). In *Proceedings of the First Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2025)*, pages 17–34, Abu Dhabi, UAE. International Committee on Computational Linguistics.
- Prajwal Thapa, Jinu Nyachhyon, Mridul Sharma, and Bal Krishna Bal. 2025. [Development of pre-trained transformer-based models for the Nepali language](#). In *Proceedings of the First Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2025)*, pages 9–16, Abu Dhabi, UAE. International Committee on Computational Linguistics.
- Bipin Timalsina, Nawaraj Paudel, and Tej Bahadur Shahi. 2022. [Attention based recurrent neural network for nepali text summarization](#). *Journal of Institute of Science and Technology*, 27(1):141–148.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Lucioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, and 374 others. 2023. [Bloom: A 176b-parameter open-access multilingual language model](#). *Preprint*, arXiv:2211.05100.
- Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. [mT5: A massively multilingual pre-trained text-to-text transformer](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. [Bertscore: Evaluating text generation with bert](#). *Preprint*, arXiv:1904.09675.

A Code and Resources

To support reproducibility and future research, the code, dataset, and models developed during this study have been made publicly available.

- **Source Code:** The code for scraping, cleaning, model training, and evaluation are available on GitHub at <https://github.com/SaskarKhadka/Nepali-News-Headline-Generation>.
- **Dataset and Models:** The dataset and trained models are available on Hugging Face Hub at <https://huggingface.co/saskarkhadka/datasets> and <https://huggingface.co/saskarkhadka/models> respectively.