From Romanized to Devanagari: Enhancing Nepali Sentiment Analysis with NepaliXlit

Suraj Patel, Kashish Kumari Dhami, Norden Sherpa, Supriya Khadka
Sunway College, Birmingham City University, Kathmandu, Nepal
{suraj_patel_a25, kashish_dhami_a25, norden_23189632, supriya}@sunway.edu.np

Abstract

Romanized Nepali is the dominant medium of social media communication in Nepal, yet most multilingual NLP models are trained on Devanagari, creating a noticeable drop in performance in informal settings. To address this script mismatch, we develop NepaliXlit, a transliteration model fine-tuned from IndicXlit to better handle the phonetic variability of Romanized Nepali. Trained on 2,943 informal word pairs and evaluated on 736 held-out pairs, NepaliXlit improves transliteration accuracy by 8% and reduces character error rate by 11%. We use sentiment analysis as a testbed to understand whether transliteration actually helps downstream NLP tasks. We curate over 6,500 Romanized social media comments and construct a balanced subset of 1,518 manually annotated instances. Baseline experiments show that multilingual encoder models struggle with Romanized input; however, transliterating text into Devanagari using NepaliXlit consistently improves sentiment classification accuracy with mBERT and MuRIL. Comparative evaluation against large language models (LLMs) further reveals that generative models such as Gemini and GPT variants exhibit strong cross-script generalization and outperform encoder-based baselines. Our results indicate that adaptive transliteration enhances conventional multilingual models, while modern LLMs offer a better alternative for multi-script, low-resource settings.

Keywords: Transliteration, Romanized Nepali, Cross-Script NLP, Sentiment Analysis

1. Introduction

The Nepali language, historically referred to as Khas Kura or Gorkhali, is the lingua franca of Nepal and is spoken by millions across South Asia and the global diaspora (Sarkar, 2026). The Devanagari script is the official writing system for Nepali and is deeply embedded in the nation's administrative and literary history (Dahal, 2000). Devanagari exhibits relatively strong grapheme-to-phoneme correspondence and forms the foundation of standard Nepali orthography (Kumar et al., 2010).

However, the digital age has introduced a significant divergence between formal written standards and informal online communication. Despite the availability of native language keyboards, the Roman script has emerged as the dominant mode of digital communication for many Indic languages, including Nepali. This shift is largely driven by the global ubiquity of QWERTY keyboards on mobile devices and computers. As noted by Madhani et al. (2023a), the learning curve associated with native keyboard layouts is often cumbersome for multilingual South Asian users, leading to a strong preference for Romanized input. This results in a vast volume of user-generated content on platforms such as Reddit, Facebook, and Twitter created using Romanized Nepali, a phonetic approximation of the language using Latin characters (Sitoula et al., 2025).

This widespread reliance on Romanized input creates a structural mismatch for Natural Language Processing (NLP). Most state-of-the-art NLP models for Nepali are pre-trained primarily on Devanagari corpora (Pudasaini et al., 2024), meaning that the script gap between user input and model training data presents a substantial barrier to effective text analysis in informal domains. While Devanagari maintains orthographic stability, Romanized Nepali is highly unstructured. Although formal Romanization systems exist, such as those documented by Shrestha (2012), they are rarely adhered to in casual social media discourse. Instead, users employ non-standardized phonetic spellings that vary heavily based on individual preference, creating noisy text that lacks consistent orthographic rules.

For example, in the sentence आज दिन राम्रो छ, meaning "Today is a good day," the word आज ($/\mathrm{a}\text{ː}.\widehat{\mathrm{dz}}\mathrm{ʌ}/$) may appear as *aaja*, *aja*, *aajaa*, or *aaj*. Similarly, छ ($/\widehat{\mathrm{ts}}^{\mathrm{h}}\mathrm{ʌ}/$) may be written as *chha*, *cha*, *chh*, *xa*, *xaa*, or simply *x*. Such variability introduces significant noise and further exacerbates the gap between real-world usage and model training distributions. In addition, the Devanagari script itself presents specific challenges and biases in NLP due to its structural complexity and the linguistic diversity of the South Asian region (Khadka and Bhattarai, 2025). Together, these factors highlight the need for approaches that explicitly address script mismatch in Nepali NLP.

To bridge this gap, transliteration, the conversion of text from one script to another based on

Proceedings of the Second workshop on Challenges in Processing South Asian Languages (CHiPSAL2026) @ LREC, pages 115–126
16 May 2026.

phonetic similarity (Madhani et al., 2023a), can be employed as a preprocessing strategy. By converting Romanized inputs into native Devanagari, noisy social media text can be normalized and made compatible with multilingual encoders and large-scale Indic language resources (Baral, 2025). However, existing transliteration tools are not tailored to the informal, phonetically inconsistent Nepali found on social media. As we demonstrate in this work, this limitation necessitates the development of a domain-adapted transliteration model capable of handling non-standard spellings.

To evaluate the efficacy of this approach, we use sentiment analysis as a simple but effective way to test downstream performance. Sentiment analysis is the computational study of people's opinions, sentiments, and emotions toward entities and events (Anderson et al., 2024). The rise of internet usage has generated a large volume of opinionated social media content, making automated sentiment detection valuable for research and industry alike (Pudasaini et al., 2024). Importantly, sentiment analysis also provides a controlled framework for assessing model performance to informal and noisy language (Barbieri et al., 2022). By applying sentiment analysis to both raw Romanized text and transliterated Devanagari text, we systematically evaluate whether bridging the script gap results in improvements in downstream NLP performance in low-resource settings.

Our core contributions are:

- We develop NepaliXlit, a domain-adapted transliteration model fine-tuned from IndicXlit to natively process informal, phonetically variable spellings.
- We compile and annotate a balanced three-class Social Sentiment Analysis dataset of Romanized Nepali social media comments.
- We demonstrate via systematic experiments with mBERT and MuRIL that transliteration consistently improves downstream classification accuracy.
- We evaluate multiple frontier Large Language Models (LLMs) against these datasets, providing a comprehensive cross-script evaluation for Nepali sentiment analysis.

2. Related Work

Prior work relevant to this study falls into four intersecting areas: Nepali transliteration systems, large-scale Indic transliteration models, sentiment analysis for Nepali and multilingual settings, and cross-script modeling challenges. Together, these strands illustrate both the progress made and the gap in handling informal Romanized Nepali.

2.1. Nepali Transliteration

The computational processing of Nepali and Devanagari scripts traditionally focuses on bridging the gap between phonetic representation and standardized digital formats. Early efforts in this domain relied heavily on rule-based systems that applied deterministic character-level mappings (Shrestha, 2012). These systems used handcrafted grapheme-to-phoneme rules to convert between scripts, often employing predefined transliteration tables and finite-state style processing to ensure consistent one-to-one correspondences (Bhalla et al., 2013; De Mel et al., 2025). Similarly, Khanal et al. (2021) utilized a manually curated mapping table to convert printed Nepali text into audible speech. While these rigid mappings work effectively for formal and predictable text, they struggle with the linguistic nuances of colloquial language.

Recent advancements have shifted toward sophisticated normalization pipelines designed to handle the inconsistencies of Romanized Nepali. The NEPTUN framework by Chaudahry et al. (2025) represents a significant step in this direction by integrating phonetic normalization with lexical validation. This system converts Romanized input into Devanagari to verify it against an established Nepali lexicon before producing a standardized Romanized output. By stabilizing the input text, this method significantly enhances the accuracy of downstream tasks such as sentiment classification. Despite these improvements, existing systems remain optimized for either formal text or lexicon-constrained normalization, and do not explicitly address the extreme phonetic variability and non-standard shorthand prevalent in informal Romanized Nepali social media.

2.2. IndicXlit and Its Limitations

The most significant advancement in Indic transliteration came through the Aksharantar project (Yash Madhani and others, 2022), which compiled the largest publicly available transliteration corpus of 26 million word pairs across 21 Indic language pairs. This dataset was used to develop IndicXlit, a multilingual Transformer-based model achieving a 15% accuracy improvement over previous evaluations. IndicXlit has since become the de facto evaluation standard in the field and infrastructure for data generation, producing lower token fertility and better embedding alignment than rule-based systems (Jaavid et al., 2024). However, because Aksharantar is constructed primarily from formal, dictionary-sourced word pairs,

IndicXlit is not optimized for the highly inconsistent spellings characteristic of informal social media input Baruah et al. (2024). Our NepaliXlit model addresses this domain gap by fine-tuning IndicXlit on social media-derived Romanized-Devanagari word pairs, specializing it for the phonetically inconsistent spellings common in casual digital discourse.

2.3. Models Used for Sentiment Analysis

Because our objective is to evaluate whether transliteration improves downstream performance, the choice of sentiment classification models must reflect current standards in Nepali NLP. While early sentiment analysis relied on probabilistic classifiers such as Naïve Bayes and SVMs (Rish et al., 2001; Forman, 2008), and later deep learning architectures such as CNN-BiLSTM hybrids (Hashmi et al., 2024), the field is now dominated by pre-trained Transformer-based models. These models consistently outperform traditional approaches in multilingual and low-resource settings, including Aspect-Based Sentiment Analysis tasks (Narayanaswamy, 2021).

In the Nepali context, Sitoula et al. (2025) demonstrated that multilingual transformers such as XLM-RoBERTa and DistilBERT outperform SVM, Random Forest, CNN, and LSTM baselines on Reddit sentiment classification. This establishes transformer-based multilingual encoders as the appropriate and competitive baseline for Nepali sentiment analysis. Building on this foundation, our work evaluates mBERT (Devlin et al., 2019) and MuRIL (Khanuja et al., 2021) to systematically assess whether transliterating Romanized input into Devanagari improves performance within this model family. We further assess frontier LLMs to examine whether large generative models exhibit stronger cross-script generalization without explicit normalization.

2.4. Cross-Script Modeling

A fundamental structural challenge in Nepali NLP lies in the mismatch between model training data and real-world usage. Most available models are optimized for Devanagari corpora or high-resource multilingual data, whereas informal digital communication is dominated by Romanized Nepali. This asymmetry creates performance degradation when models trained on standardized script distributions encounter phonetically variable Romanized input. Large multilingual encoders are further affected by the "curse of multilinguality" (Xu et al., 2021), where model capacity is diluted across many languages, disproportionately impacting low-resource ones such as Nepali. Within the Nepali ecosystem, NepBERTa (Timilsina et al., 2022) represents a strong monolingual Devanagari model and has shown effectiveness on YouTube comments (Pudasaini et al., 2024). However, it was not designed to process Romanized input. Similarly, NepaliGPT (Pudasaini et al., 2025) was trained on a strictly filtered Devanagari corpus that excluded non-Devanagari tokens, rendering it unsuitable for Romanized or code-mixed settings.

Some work has explored Romanized Nepali directly. For example, Pradhananga and Sah (2023) demonstrated that BERT outperforms traditional classifiers on Romanized reviews. However, these approaches do not address the structural script mismatch between training corpora and informal usage. Rather than training separate models per script, our work investigates whether domain-adapted transliteration can bridge this divide and restore compatibility with existing Devanagari-trained encoders, while also comparing performance against modern LLMs that may inherently generalize across scripts.

3. Methodology

This section outlines our three-phase methodology for sentiment analysis of Romanized Nepali text. First, we detail the architecture and fine-tuning of our transliteration model, NepaliXlit. Next, we describe the dataset creation and annotation process for social media comments. Finally, we present the experimental setup used to evaluate model performance across different script configurations.

3.1. Development of NepaliXlit

The core challenge in processing Nepali social media text is the script gap between informal Romanized input and models trained on Devanagari (Pudasaini et al., 2024). Without normalization, Transformer-based classifiers encounter highly variable Romanized forms, resulting in degraded performance. To address this, we develop *NepaliXlit*, a domain-adapted transliteration model that converts noisy Romanized strings into standardized Devanagari[1]. Following the transliteration-as-normalization paradigm validated by Sampath and Supriya (2024), NepaliXlit is specifically fine-tuned to handle the high phonetic variability of informal "Internet Nepali," mapping non-standard strings to their correct linguistic counterparts. Example mappings are shown in Table 1.

By normalizing these variations at the preprocessing stage, text becomes compatible with downstream Devanagari-trained models such as

[1]NepaliXlit is publicly available on https://github.com/Supriya090/NepaliXlit/tree/master/cli

Romanized	Devanagari	Phonetic (IPA)	English
manxe	मान्छे	/maːn.t͡sʰ e/	person
hunxa	हुन्छ	/hun.t͡sʰ ʌ/	okay / happens
garnaw	गर्न	/gʌr.nʌ/	to do

Table 1: Sample transliteration word mappings from the NepaliXlit training dataset.

Romanized Input	Transliterated Results	English Translation
Priyanka le kati ramro boleko k	प्रियंका ले कति राम्रो बोलेको के	Priyanka speaks so well.
Rekha thapa mam lai leuna paro	रेखा थापा माम लाई ल्युन परो	We need to bring Rekha Thapa Ma'am.
mahina vaisakyo ghar dekhi baahira gako xaina	महिना भइसक्यो घर देखि बाहिर गाको छैन	It's been months since I left the house.
Nepal ma kam garni manche na uha parnu hunxa	नेपाल मा काम गर्नी मान्छे न उहाँ पर्नु हुन्छ	He is the one who works in Nepal.
Priyanka kati choti yo maan jetcheu Mero	प्रियंका कति छोटी यो मान जेट्छेउ मेरो	*Priyanka, how many times will you win my heart?*

Table 2: Qualitative examples of transliteration generated using NepaliXlit. The first four samples demonstrate accurate conversion of informal social media text, while the final sample (italicized) highlights a model error where "jetcheu" was incorrectly mapped to "जेट्छेउ" instead of the contextually appropriate "जित्छौ".

mBERT (Devlin et al., 2019) and MuRIL (Khanuja et al., 2021).

3.1.1. Architecture and Corpus Preparation

We selected IndicXlit (Madhani et al., 2023b) as the base architecture due to its transformer-based encoder-decoder design and native support for Nepali script. To adapt it for informal social media text, we employed a domain-adaptation strategy via fine-tuning, retaining general Indic transliteration capabilities while accommodating the orthographic variability in our dataset (Baruah et al., 2024). The fine-tuning corpus consists of 3,679 parallel Romanized–Devanagari word pairs, sourced from 2,500 social media comments collected across Facebook and Reddit. We prioritized high-frequency terms with observed informal spellings. Each mapping was manually verified by two native Nepali speakers to ensure correctness. Character-level tokenization was applied, with space-separated characters, allowing the model to capture diverse spelling variations without relying on a fixed vocabulary. We partitioned the dataset into 80% training and 20% testing splits. The held-out test set contains informal spellings representative of real-world social media usage, allowing us to directly evaluate domain-specific transliteration performance.

3.1.2. Fine-tuning and Evaluation

We fine-tuned the model using the Fairseq framework (Ott et al., 2019), retaining the sequence-to-sequence Transformer encoder-decoder architecture of IndicXlit. NepaliXlit was trained on the 80% training portion of the Romanized–Devanagari corpus and evaluated on the held-out 20% test split consisting of informal social media word forms. Training was conducted on an NVIDIA GeForce RTX 3060 (8GB).

We evaluated transliteration quality using Top-1 Accuracy and Character Error Rate (CER). Compared to the base IndicXlit model evaluated on the same test set, NepaliXlit achieved a **relative improvement of approximately 8% in Top-1 Accuracy** and a **relative reduction of 11% in CER**. These improvements indicate that domain-specific fine-tuning enhances robustness to noisy and phonetically variable Romanized Nepali text. While this work focuses primarily on the downstream impact of transliteration on sentiment classification, these results confirm that NepaliXlit provides a stronger normalization layer than the base IndicXlit model. Some examples of the transliteration done by NepaliXlit is given in Table 2. A more extensive re-evaluation of transliteration performance with additional metrics and larger evaluation sets is left for future work.

3.2. Sentiment Dataset Creation

3.2.1. Data Collection and Filtering

We collected Romanized Nepali comments from multiple social media platforms, primarily targeting Facebook, followed by YouTube and Instagram. To ensure a high density of Romanized

Nepali text, we specifically focused on content popular among youth audiences. On Facebook, we scraped comments from high-engagement pages such as *Routine of Nepal Banda*[2]. On YouTube, we targeted popular podcasts and vlogs, selecting channels such as *Surakshya KC*[3] and *Priyanka Karki*[4], where viewers frequently interact using Romanized script.

Using the YouTube Comment Exporter extension[5], along with manual extraction for Facebook and Instagram, we conducted 69 collection runs across 69 distinct posts. Although the exporter has a maximum capacity of 100 comments per post, we retrieved an average of 80 comments per run to maintain data quality, resulting in an initial pool of 6,566 raw comments. The raw dataset consisted of a mix of emojis, pure English text, Devanagari Nepali, and multilingual content. We utilized Gemini Pro (Comanici et al., 2025) to filter out non-target content, specifically removing emojis, English-only sentences, and Devanagari text to isolate a cleaned Romanized Nepali dataset.

3.2.2. Sentiment Annotation and Balancing

We manually labeled each comment as Neutral (0), Positive (1), or Negative (2), following standard three-class sentiment analysis practices (Anderson et al., 2024). The annotation was conducted by two students who are active social media users, ensuring they were well-versed in the slang and nuances of modern Romanized Nepali.

Sentiment	Initial Count	Final Count
Neutral	582	515
Positive	510	501
Negative	409	502

Table 3: Sentiment class distribution before and after balancing.

As shown in Table 3, the initial filtering of the 6,500+ comments resulted in a distribution where the negative class was significantly underrepresented (409 samples) compared to neutral and positive classes. To address this imbalance and prevent model bias (Sitoula et al., 2025), we performed a targeted second round of scraping. During this phase, we specifically sought out posts with controversial topics or heated discussions likely to yield negative sentiments. By repeating the collection and cleaning process for these specific sources, we successfully increased the negative sample size to match the other classes. The finalized, balanced dataset consists of 1,518 comments, providing a foundation for evaluating transformer-based and large language models[6].

3.3. Experimental Setup

We designed experiments to compare sentiment analysis performance across Romanized Nepali, Devanagari Nepali, and combined script datasets. We evaluated both fine-tuned Transformer models and LLMs for this comparison.

3.3.1. Transformer Models and Training Configuration

We fine-tuned two Transformer-based encoders, selected for their strengths in multilingual and Indic-specific contexts. Specifically, we used mBERT, pre-trained on 104 languages using masked language modeling (Devlin et al., 2019), and MuRIL, pre-trained on 17 Indic languages along with their transliterated counterparts (Khanuja et al., 2021). MuRIL's architecture is optimized for Devanagari text, making it suitable for transliterated datasets.

All models were trained on datasets consisting of text paired with corresponding sentiment labels. We enforced strict train–test separation to prevent data leakage. Notably, this test set differs from the "held-out" test set used for finetuning IndicXlit. MuRIL was trained exclusively on transliterated Devanagari data, while mBERT was evaluated under three configurations: Romanized only, Devanagari only, and a combination of both scripts (Table 4). All models were trained on Google Colab using GPU acceleration.

Configuration	Train	Test
Romanized only	1200	318
Devanagari only	1200	318
Combined scripts	2400	636

Table 4: Fine-tuning configurations for sentiment classification.

3.3.2. Large Language Models Evaluation

We selected LLMs via OpenRouter[7] for multilingual support and efficiency. We evaluated them

[2] https://www.facebook.com/officialroutineofnepalbanda/

[3] https://www.youtube.com/@SURAKSHYAKCOFFICIAL

[4] https://www.youtube.com/@PriyankaKarki11

[5] https://chromewebstore.google.com/detail/youtube-comment-exporter/hfkfcomfcfeanhckcdohojjjlaagmgpd

[6] Social Sentiment Analysis Dataset is publicly available at https://github.com/anynomousresearch/Data

[7] https://openrouter.ai/

in a zero-shot setting using the following prompt:

> Classify the sentiment of this Nepali comment: {comment}
> Labels: 0 Neutral; 1 Positive; 2 Negative.
> Answer with only the label.

We included Gemini 2.5 Pro (Google DeepMind) (Comanici et al., 2025) for its advanced handling of mixed scripts, effective with both Romanized and Devanagari text. We used LLaMA 3.3 70B Instruct (Meta) (Grattafiori et al., 2024), an open-source instruction-tuned model capable of multilingual tasks, and Qwen3 Max Thinking (Alibaba) (Yang et al., 2025), optimized for reasoning and multilingual processing. Claude Haiku 4.5 (Anthropic) served as a low-cost model suitable for short classification tasks, while Grok 4.1 Fast (xAI) provided speed and efficiency for bulk processing. We further incorporated GPT-OSS 120B (OpenAI) (Agarwal et al., 2025) as an open-source baseline and GPT-5.2 (OpenAI) as a high-performance model for sentiment analysis. This careful selection balanced high-accuracy with cost-efficient bulk classification.

3.3.3. Experimental Workflow

We designed a four-stage pipeline to ensure systematic evaluation and reproducibility. First, we applied Script Standardization, using NepaliXlit to transliterate Romanized comments into Devanagari, which created parallel datasets to minimize orthographic noise. Second, during the Dataset Configuration stage, we organized these outputs into Romanized, Devanagari, and combined subsets with fixed train-test splits. Third, we conducted Transformer Fine-Tuning on mBERT and MuRIL using consistent hyperparameters, ensuring a fair comparison across scripts. Finally, we executed the LLM Evaluation stage, utilizing an asynchronous zero-shot protocol to evaluate frontier models. We employed concurrency controls and batched processing in this final stage, ensuring stable and scalable inference across all dataset variations.

4. Results

This section presents the results we obtained from the experimental configurations described in Section 3. We evaluated performance using overall accuracy as the primary metric, maintaining consistency with prior Nepali sentiment analysis evaluation studies (Sitoula et al., 2025; Pudasaini et al., 2024). We report the results separately for encoder-based models and LLMs.

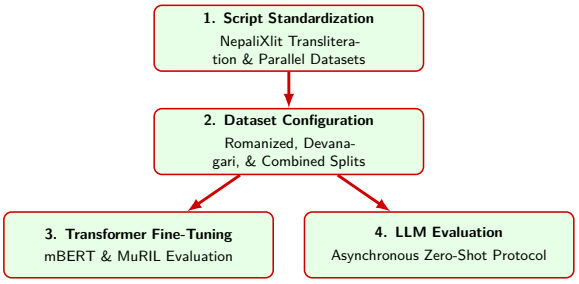

Figure 1: Four-stage experimental workflow. The pipeline sequentially advances from script standardization and dataset configuration to transformer fine-tuning and LLM evaluation phases.

Model	Input Script	Baseline (%)	Finetuned (%)
mBERT	Romanized	35.85	**49.69**
mBERT	Devanagari	34.91	**53.77**
mBERT	Combined	36.16	**50.00**
MuRIL	Devanagari	31.76	**60.06**

Table 5: Comparison of baseline and fine-tuned model accuracy for the sentiment analysis task across different input scripts. Bold values indicate the performance improvements after fine-tuning.

4.1. Impact of Script Normalisation on Encoder-Based Model Performance

Table 5 presents a comprehensive comparison of encoder-based model performance across different input scripts and normalisation strategies. In the baseline configuration lacking script normalisation, both mBERT and MuRIL exhibited notably low accuracy on Romanized input, scoring 35.85% and 31.76% respectively. These results highlight the significant difficulty these models face in processing informal Romanized Nepali text.

We evaluated NepaliXlit on a held-out transliteration test set, where it increased Top-1 accuracy by 8% and reduced character error rate by approximately 11% compared to the IndicXlit baseline. Applying this transliteration as a preprocessing step created standardized Devanagari input, which improved encoder performance across all configurations. Notably, the Devanagari-only transliterated configuration for mBERT achieved higher accuracy than the Romanized-only setup, confirming the benefits of script standardisation.

MuRIL achieved the strongest overall performance among encoder models, reaching 60.06% accuracy on the transliterated dataset. Despite these overall improvements, classifying the neutral sentiment class remained the most significant challenge across all configurations, as detailed in Table 6.

Model	Training Configuration	Overall (%)	Neutral (%)	Positive (%)	Negative (%)
mBERT	Romanized-only	49.69	40.87	51.49	57.84
mBERT	Devanagari-only (Translit.)	53.77	50.43	51.49	59.80
mBERT	Combined (Roman. + Deva.)	50.00	33.04	65.35	53.92
MuRIL	Devanagari (Translit.)	60.06	38.26	74.26	70.59

Table 6: Accuracy performance comparison after transliteration-based fine-tuning. We can see that neutral is consistently the worst performing label (denoted by an underline).

Model	Romanized				Devanagari			
	Overall (%)	Neu. (%)	Pos. (%)	Neg. (%)	Overall (%)	Neu. (%)	Pos. (%)	Neg. (%)
Gemini-2.5-Pro	**73.14**	37.39	**97.03**	91.40	68.27	31.30	**92.93**	86.73
GPT-5.2	72.06	30.36	96.04	**94.12**	**69.52**	34.78	90.10	**88.89**
GPT-OSS-120B	63.92	**38.94**	84.16	71.57	62.10	**40.87**	75.26	73.53
Grok-4.1-Fast	61.95	20.00	83.17	88.24	60.26	19.13	80.00	88.24
Qwen3-Max	60.58	22.61	85.26	80.39	60.90	28.44	81.19	75.49

Table 7: Sentiment analysis performance across scripts (bold indicates highest value per column).

4.2. LLM Performance Results

Table 7 presents the zero-shot evaluation results we obtained by evaluating multiple large language models on Romanized and transliterated Devanagari datasets. Gemini-2.5-Pro achieved the highest overall accuracy of 73.14% on Romanized input, and GPT-5.2 followed closely at 72.06%. Furthermore, GPT-5.2 demonstrated strong cross-script consistency by maintaining a competitive performance of 69.52% on Devanagari input. This strong cross-script performance of frontier LLMs supports the broader findings of Azam et al. (2025), who concluded that large general-purpose language models exhibit stronger cross-script generalization than specialized fine-tuned models for Indic languages. The persistent difficulty with neutral sentiment classification across all LLMs mirrors the pattern we observed in encoder-based models and aligns directly with the research of Sitoula et al. (2025).

4.3. Comparative Analysis

Our results reveal that LLMs substantially outperform encoder-based models in overall accuracy for Nepali sentiment analysis. Furthermore, a clear divergence in optimal preprocessing strategies emerged between the two model families. While transliteration to Devanagari proved essential for maximizing the performance of encoder-based models like mBERT and MuRIL, frontier LLMs exhibited superior performance when processing the original Romanized input.

As illustrated in Figure 2, the highest-performing models Gemini-2.5-Pro and GPT-5.2 achieved their peak accuracy on Romanized text, experiencing a performance degradation when evaluated on transliterated Devanagari. These findings indicate that while adaptive transliteration remains a critical bridge for conventional multilingual encoders (Madhani et al., 2023b), it may be less critical for modern generative models that demonstrate stronger cross-script generalization (Azam et al., 2025). This divergence suggests that the "script gap" affects discriminative and generative architectures differently, a trend we explore further in the Discussion.

Figure 2: Zero-shot performance of LLMs across Romanized and Devanagari datasets. Gemini-2.5-Pro and GPT-5.2 outperform other models; most LLMs show higher accuracy on Romanized input.

5. Discussion

Our experimental results offer meaningful insights into handling informal, Romanized Nepali within modern NLP pipelines. Our findings validate the hypothesis that transliteration remains a vital preprocessing step for encoder-based models. Simultaneously, we reveal that state-of-the-art LLMs

possess an inherent adaptability to script variation that challenges traditional adaptation strategies. This section discusses the implications of these findings within the context of existing literature and identifies key areas for future research.

5.1. The Efficacy of Domain-Adapted Transliteration

The significant performance gain observed in encoder-based models following transliteration underscores the persistence of the script gap in low-resource languages. The difficulty standard multilingual encoders face with raw Romanized input suggests a lack of sufficient exposure to informal Romanized orthographies during pre-training, likely due to the historical dominance of Devanagari script in their training corpora (Khanuja et al., 2021). Furthermore, non-standard spellings typical of social media platforms cause severe token fragmentation. In these instances, the model fails to recognize informal Romanized words as Nepali, treating them instead as out-of-vocabulary or noise (Rust et al., 2021). By utilizing NepaliXlit to convert text into standardized Devanagari, we allow the model to operate within its intended linguistic and structural environment, leading to the observed jump in accuracy. These results extend the arguments of Madhani et al. (2023b), who posited that effective transliteration tools are a prerequisite for Indic NLP. However, our findings specifically demonstrate that generalist models like IndicXlit are often insufficient for the "noisy" social media domain. Consistent with Baruah et al. (2024), our success confirms that effective domain adaptation requires more than simple script conversion; it requires the normalization of sociolect-driven spellings that characterize modern digital communication in Nepal (Sitoula et al., 2025).

5.2. The Shift from Fine-Tuning to Zero-Shot LLMs

The fact that Gemini-2.5-Pro outperformed the best fine-tuned encoder (MuRIL) by over 13 percentage points without task-specific training signals a paradigm shift for low-resource language tasks. This outcome supports the broader observation by Azam et al. (2025) that large general-purpose language models exhibit stronger cross-script generalization than specialized fine-tuned models.

Building upon our analysis in Section 4.3, we attribute the LLMs' native proficiency with Romanized text to three primary factors. First, internet-scale LLMs are extensively pre-trained on informal social media data (Shi et al., 2023), making them inherently robust to Romanized Indic text compared to encoders trained on formal Wikipedia corpora. Second, modern LLM tokenizers are heavily biased toward Latin scripts; in contrast, Devanagari script often suffers from token fragmentation, which can degrade semantic processing (Kanjirangat et al., 2025). Finally, evaluating LLMs directly on Romanized text avoids the error propagation inherent in any intermediate transliteration pipeline.

However, it is important to qualify these findings to avoid overgeneralization. While transliteration appears redundant for high-capacity frontier models in this informal domain, it remains a vital bridge for deploying specialized encoder models (Sampath and Supriya, 2024). Smaller models like MuRIL and mBERT remain necessary for tasks requiring high speed and lower costs, especially in localized technology projects in Nepal where hardware resources are limited (Schwartz et al., 2020). Therefore, while frontier LLMs offer a streamlined, preprocessing-free alternative, domain-adapted tools like NepaliXlit remain essential for maintaining a sustainable and diverse NLP ecosystem.

5.3. The Ambiguity of Neutral Sentiment in Low-Resource Contexts

Across all model architectures and script configurations, the neutral sentiment class consistently presented the greatest classification challenge. This systematic difficulty mirrors the findings of Sitoula et al. (2025), who observed similar ambiguity in Nepali Reddit data. We attribute this to the linguistic nature of neutral social media comments, which often lack explicit sentiment-bearing lexicons and rely heavily on context-dependent interpretation. Current models fail to capture the pragmatic cues that differentiate informational statements from mildly positive or negative expressions in Nepali. We suggest that future annotation efforts prioritize inter-annotator agreement metrics, such as Cohen's Kappa, to better quantify the inherent subjectivity of the neutral class (Anderson et al., 2024). Furthermore, researchers might yield more reliable outcomes for neutral instances by adopting a more granular annotation scheme or utilizing emotion detection rather than broad sentiment polarity (Pudasaini et al., 2024).

6. Limitations and Future Directions

While this study establishes a new baseline for Romanized Nepali sentiment analysis, several limitations remain. First, the dataset size of 1,518 annotated comments, though balanced, is relatively small for training deep transformer models from scratch, which typically require large-scale data to generalize effectively and avoid overfitting (Devlin et al., 2019). Second, our evaluation

of LLMs was limited to zero-shot prompting; incorporating few-shot or chain-of-thought prompting could better leverage the reasoning capabilities of advanced models such as GPT-5.2 and Qwen3-Max (Azam et al., 2025). Third, the study focuses exclusively on monolingual Romanized and Devanagari Nepali text, explicitly excluding code-mixed Nepali-English data during preprocessing. While this ensured a controlled experimental setup for analyzing transliteration, it limits ecological validity, as code-mixing is highly prevalent in real-world Nepali digital communication. Additionally, we did not empirically measure token fragmentation across scripts, instead relying on assumptions from prior studies that Devanagari text may exhibit higher fragmentation; a systematic analysis could reveal important insights into model performance differences.

Future work should address these limitations by scaling the dataset, incorporating more advanced prompting strategies, and extending the framework to handle code-mixed Nepali-English text. Furthermore, evaluating tokenization behavior across scripts and expanding the pipeline to other downstream tasks such as hate speech detection (Khadka et al., 2025), named entity recognition and text-to-speech (Khadka et al., 2023) would help validate the broader applicability of the NepaliXlit tool in real-world scenarios.

7. Conclusion

This paper addresses the challenge of sentiment analysis on informal Romanized Nepali social media text. We introduce a balanced dataset of 1,518 annotated comments and develop NepaliXlit, a domain-adapted transliteration model fine-tuned for sociolect-driven spellings. Our evaluation demonstrates that bridging the script gap through transliteration consistently improves encoder-based sentiment classification. Specifically, MuRIL achieved 60.06% accuracy on transliterated Devanagari data. However, comprehensive zero-shot evaluation reveals that frontier LLMs substantially outperform fine-tuned encoders directly on Romanized input. Notably, Gemini-2.5-Pro reached 73.14% accuracy, suggesting inherently adaptable cross-script representations without task-specific adaptation. Despite these advancements, accurately classifying neutral sentiment remains a persistent challenge across all model architectures. Ultimately, specialized tools like NepaliXlit fill a critical gap for low-resource encoder pipelines. Yet, the strong performance of frontier LLMs indicates a shift in how systems will process informal digital communication in the future. The dataset, model, and evaluation results we present in this work serve as a foundational resource for advancing multilingual NLP and cross-script sentiment analysis in underserved contexts.

8. Ethical Considerations

This study was conducted with strict adherence to ethical standards regarding data privacy and research integrity. The dataset consists of publicly available social media comments, and to ensure the anonymity of users, all personally identifiable information, such as usernames, profile links, and specific mentions, was removed during the preprocessing phase. Furthermore, the manual annotation process was conducted objectively to minimize demographic or linguistic bias. The fine-tuning of the NepaliXlit model and the analysis of LLMs were performed for academic research purposes, ensuring that the results are used to improve linguistic accessibility for low-resource languages without promoting harmful or biased sentiment classification.

9. Data/Code Availability Statement

All the data and code supporting the findings of this study is publicly available on GitHub.

10. References

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. 2025. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*.

Tess Anderson, Sayani Sarkar, and Robert Kelley. 2024. Analyzing public sentiment on sustainability: A comprehensive review and application of sentiment analysis techniques. *Natural Language Processing Journal*, 8:100097.

Gulfarogh Azam, Mohd Sadique, Saif Ali, Mohammad Nadeem, Erik Cambria, Shahab Saquib Sohail, and Mohammad Sultan Alam. 2025. Beyond specialization: Benchmarking llms for transliteration of indian languages. *arXiv preprint arXiv:2505.19851*.

Dipesh Baral. 2025. Comparing labse with contrastively and soft-label fine-tuned mbert models for semantic search over a nepali knowledge base. *International Journal on Engineering Technology*, 3(1):146–155.

Francesco Barbieri, Luis Espinosa Anke, and Jose Camacho-Collados. 2022. Xlm-t: Multilingual

language models in twitter for sentiment analysis and beyond. In *Proceedings of the thirteenth language resources and evaluation conference*, pages 258–266.

Hemanta Baruah, Sanasam Ranbir Singh, and Priyankoo Sarmah. 2024. Assamesebacktranslit: Back transliteration of romanized assamese social media text. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 1627–1637.

Deepti Bhalla, Nisheeth Joshi, and Iti Mathur. 2013. Rule based transliteration scheme for english to punjabi. *arXiv preprint arXiv:1307.4300*.

Chandra Prakash Chaudahry, Basanta Joshi, Aman Shakya, and Santosh Giri. 2025. Neptun: Normalization for romanized nepali sentiment analysis. *Journal of Himalaya College of Engineering*, 2(1).

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Ram Kumar Dahal. 2000. Language polics innepal. *Contributions to Nepalese Studies*, 27(2):155–190.

Yomal De Mel, Kasun Wickramasinghe, Nisansa De Silva, and Surangika Ranathunga. 2025. Sinhala transliteration: a comparative analysis between rule-based and seq2seq approaches. In *Proceedings of the First Workshop on Natural Language Processing for Indo-Aryan and Dravidian Languages*, pages 166–173.

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.

George Forman. 2008. Bns feature scaling: an improved representation over tf-idf for svm text classification. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 263–270.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Ehtesham Hashmi, Sule Yildirim Yayilgan, Ibrahim A Hameed, Muhammad Mudassar Yamin, Mohib Ullah, and Mohamed Abomhara. 2024. Enhancing multilingual hate speech detection: From language-specific insights to cross-linguistic integration. *IEEE Access*, 12:121507–121537.

J Jaavid, Raj Dabre, M Aswanth, Jay Gala, Thanmay Jayakumar, Ratish Puduppully, and Anoop Kunchukuttan. 2024. Romansetu: Efficiently unlocking multilingual capabilities of large language models via romanization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15593–15615.

Vani Kanjirangat, Tanja Samardzic, Ljiljana Dolamic, and Fabio Rinaldi. 2025. Tokenization and representation biases in multilingual models on dialectal nlp tasks. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 24003–24021.

Pilot Khadka, Ankit Bk, Ashish Acharya, Bikram Kc, Sandesh Shrestha, and Rabin Thapa. 2025. Nepali transformers@ nlu of devanagari script languages 2025: Detection of language, hate speech and targets. In *Proceedings of the First Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2025)*, pages 314–319.

Supriya Khadka and Bijayan Bhattarai. 2025. Gender bias in nepali-english machine translation: A comparison of llms and existing mt systems. In *Proceedings of the 6th Workshop on Gender Bias in Natural Language Processing (GeBNLP)*, pages 75–82.

Supriya Khadka, Ranju G.C., Prabin Paudel, Rahul Shah, and Basanta Joshi. 2023. Nepali text-to-speech synthesis using tacotron2 for mel-spectrogram generation. In *SIGUL 2023, 2nd Annual Meeting of the Special Interest Group on Under-resourced Languages: a Satellite Workshop of Interspeech 2023*.

Samir Khanal, Ranjan Paudel, and Rajendra Chataut. 2021. Nepali printed text to speech with accurate transliterated form. *International Journal of Computer Science and Information Technologies (IJCSIT)*, 12(5):97–101.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, et al. 2021. Muril: Multilingual representations for indian languages. *arXiv preprint arXiv:2103.10730*.

Uttam Kumar, Tanusree Das, Raju S Bapi, Prakash Padakannaya, R Malatesha Joshi, and Nandini C Singh. 2010. Reading different orthographies: an fmri study of phrase reading in hindi–english bilinguals. *Reading and Writing*, 23(2):239–255.

Yash Madhani, Mitesh M Khapra, and Anoop Kunchukuttan. 2023a. Bhasa-abhijnaanam: Native-script and romanized language identification for 22 indic languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 816–826.

Yash Madhani, Sushane Parthan, Priyanka Bedekar, Gokul Nc, Ruchi Khapra, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh M Khapra. 2023b. Aksharantar: Open indic-language transliteration datasets and models for the next billion users. In *Findings of the association for computational linguistics: Emnlp 2023*, pages 40–57.

Gagan Reddy Narayanaswamy. 2021. *Exploiting BERT and RoBERTa to Improve Performance for Aspect Based Sentiment Analysis*. Dissertation, Technological University Dublin, Dublin, Ireland.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics (Demonstrations)*, pages 48–53.

Abhash Pradhananga and Anand Kumar Sah. 2023. Transformer-based deep learning models for sentiment analysis in romanized nepali: a comparative investigation of bert and roberta. In *Proceedings of 14th IOE graduate conference*, volume 14.

Shushanta Pudasaini, Sunil Ghimire, Prabhat Ale, Aman Shakya, Prakriti Paudel, and Basanta Joshi. 2024. Application of nepali large language models to improve sentiment analysis. In *Proceedings of the 2024 7th International Conference on Computers in Management and Business*, pages 144–150.

Shushanta Pudasaini, Aman Shakya, Siddhartha Shrestha, Sahil Bhatta, Sunil Thapa, and Sushmita Palikhe. 2025. Nepaligpt: A generative language model for the nepali language. *arXiv preprint arXiv:2506.16399*.

Irina Rish et al. 2001. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46. Seattle, USA.

Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. 2021. How good is your tokenizer? on the monolingual performance of multilingual language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3118–3135.

Koyyalagunta Krishna Sampath and M Supriya. 2024. Transformer based sentiment analysis on code mixed data. *Procedia Computer Science*, 233:682–691.

Pintu Sarkar. 2026. The evolution of the nepali language: A brief history. *International Journal of Creative Research Thoughts (IJCRT)*, 14(1):a874–a880. Available at: http://www.ijcrt.org/papers/IJCRT2601109.pdf.

Roy Schwartz, Jesse Dodge, Noah A Smith, and Oren Etzioni. 2020. Green ai. *Communications of the ACM*, 63(12):54–63.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. 2023. Detecting pretraining data from large language models. *arXiv preprint arXiv:2310.16789*.

Suresh Man Shrestha. 2012. Transliteration system for nepali language. *Journal on Geoinformatics, Nepal*, 11:37–41.

Sameer Sitoula, Tej Bahadur Shahi, Laxmi Prasad Bhatt, Anisha Pokhrel, and Arjun Neupane. 2025. Nepemo: A multi-label emotion and sentiment analysis on nepali reddit with linguistic insights and temporal trends. *arXiv preprint arXiv:2512.22823*.

Sulav Timilsina, Milan Gautam, and Binod Bhattarai. 2022. Nepberta: Nepali language model trained in a large corpus. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 273–284.

Haoran Xu, Benjamin Van Durme, and Kenton Murray. 2021. Bert, mbert, or bibert? a study on contextualized embeddings for neural machine

translation. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 6663–6675.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

11. Language Resource References

Yash Madhani and others. 2022. *Aksharantar: Open Transliteration Tools for the Next Billion Users*. arXiv. Language Resource.