AssameseBackTranslit: Back Transliteration of Romanized Assamese Social Media Text

Hemanta Baruah, Sanasam Ranbir Singh, Priyankoo Sarmah
Indian Institute of Technology Guwahati
North Guwahati, India
{hemanta.b, ranbir, priyankoo}@iitg.ac.in

Abstract

This paper presents a novel back transliteration dataset capturing native language text originally composed in the Roman/Latin script, harvested from popular social media platforms, along with its corresponding representation in the native Assamese script. Assamese, categorized as a low-resource language within the Indo-Aryan language family, predominantly spoken in the north-east Indian state of Assam, faces a scarcity of linguistic resources. The dataset comprises a total of 60,312 Roman-native parallel transliterated sentences. This paper diverges from conventional forward transliteration datasets consisting mainly of named entities and technical terms, instead presenting a novel transliteration dataset cultivated from three prominent social media platforms, Facebook, Twitter (currently X), and YouTube, in the backward transliteration direction. The paper offers a comprehensive examination of ten state-of-the-art word-level transliteration models within the context of this dataset, encompassing transliteration evaluation benchmarks, extensive performance assessments, and a discussion of the unique challenges encountered during the processing of transliterated social media content. Our approach involves the initial use of two statistical transliteration models, followed by the training of two state-of-the-art neural network-based transliteration models, evaluation of three publicly available pre-trained models, and ultimately fine-tuning one existing state-of-the-art multilingual transliteration model along with two pre-trained large language models using the collected datasets. Notably, the Neural Transformer model outperforms all other baseline transliteration models, achieving the lowest Word Error Rate (WER) and Character Error Rate (CER), and the highest BLEU (up to 4 gram) score of 55.05, 19.44, and 69.15, respectively.

Keywords: Language Resource, Social Media, Transliteration, Romanization

1. Introduction

Transliteration, the process of converting text from one script into another, plays a crucial role in bridging language barriers and facilitating cross-lingual communication. While significant efforts have been made to develop transliteration datasets for various language pairs, many low-resource languages still lack comprehensive resources. Assamese, belonging to the Indo-Aryan language family and primarily spoken in the northeastern state of Assam in India, falls into this category of languages with limited transliteration datasets. In this paper, we address this gap by introducing a novel and extensive transliteration dataset for Assamese extracted from contemporary social media platforms. The choice of the social media domain is motivated by the fact that India, being a multilingual country, boasts a significant number of bilingual social media users. According to some earlier reports from KPMG[1] and Statista[2], the number of users contributing Indian language content to the Internet is growing rapidly, expected to represent nearly 75% of the total Indian Internet user base in 2021. This growth is largely attributed to the convenience of writing Indic languages using the Roman script, facilitated by English QWERTY keyboards. However, the lack of processing tools for such texts, they often go unnoticed.

Existing transliteration datasets primarily focus on named entities and out-of-vocabulary (OOV) terms, often neglecting the wealth of linguistic diversity found in user-generated content on platforms such as Facebook, Twitter, and YouTube. These social media platforms are known for hosting a myriad of content where native-language words are frequently represented in transliterated form. It is also important to note that most existing transliteration datasets predominantly focus on forward transliteration, which involves the careful transliteration of native-language words into non-native scripts with the assistance of human annotators. In contrast, our dataset emphasizes back transliteration, which involves the conversion of transliterated text written in a non-native script back to its corresponding text in the native script (Knight and Graehl, 1997). Back transliteration datasets are significant because they allow transliteration variations, offering a more flexible approach. In the forward transliteration direction, annotators tend to perform precise annotation, which often lacks the transliteration variations observed in back transliteration datasets.

In majority of the studies, while generating the forward transliterated dataset, native language script texts are primarily sourced from Wikipedia.

[1] KPMG
[2] Statista

LREC-COLING 2024, pages 1627–1637

Annotators are tasked with producing the Romanized version of these native texts, relying on their own phonetic judgment since there are no established transliteration or romanization guidelines present for most of the indic languages. For instance, the Assamese term "শুভেচ্ছা", pronunciation: /xub^{h}essɑ/, meaning: "good wishes", only three different transliteration variations: "xubhessa", "xubhescha" and "shubhessa" as adopted by the annotators observed in the forward transliterated Aksharantar dataset (Madhani et al., 2022). In contrast, our collected dataset reveals a total of 46 different variations for the same word "শুভেচ্ছা" written in Roman script, such as "hubhesaa", "xuvesa", "huvasha", "huvesha", "subhecha", "huveshya", "subhesa", "xuvesha", "xuvessa", "huvasa", "khuvessa", "kuvessa", "xubheisha", "khuveswa" and "shubhessa", among others, reflecting real instances from diverse users across three different social media platforms. In our previous studies (Baruah et al., 2024, 2023), details of the transliteration variations and challenges are reported. This demonstrates the scale and challenges associated with our dataset compared to traditional forward transliterated datasets. In the preparation of backward transliteration dataset for Assamese, users are free to express their native words using Roman script, and annotators are asked to identify the correct native word based on the context and transliterate it back to the native word in Assamese script, resulting in a richer and more diverse dataset. Our dataset aims to capture this linguistic richness and diversity, offering a valuable resource for transliteration research in the context of Assamese language.

In addition to introducing the dataset, we assess various state-of-the-art transliteration models. We initially employ two statistical baseline models of NEWS 2018 (Singhania et al., 2018), namely, a joint n-gram-based string transduction system, SEQUITUR[3] (Bisani and Ney, 2008) and a phrase-based statistical transliteration model using Moses[4] (Koehn et al., 2007) decoder. Subsequently, we transition to advanced deep neural network models, including BiLSTM with attention (Bahdanau et al., 2015) and the neural transformer (Vaswani et al., 2017) model. Additionally, we report the results obtained from three publicly available transliteration APIs: indictrans (Bhat et al., 2014), google transliteration API for the Google Input Tools[5], and a multilingual transliteration model, IndicXlit (Madhani et al., 2022). Finally, we fine-tuned three state-of-the-art pre-trained models: the multilingual IndicXlit model, Google's multilingual text-to-text transformer-based large language models, mT5 (Xue et al., 2021b), and a tokenizer-free extension of the mT5 model, ByT5 (Xue et al., 2021a).

In summary, this paper not only introduces a unique and expansive backward transliteration dataset in Assamese but also contributes to the advancement of transliteration technology by evaluating the performance of cutting-edge models on this resource. These efforts collectively aim to enhance cross-lingual communication and promote the development of linguistic resources for low-resource languages like Assamese. To the best of our knowledge, this dataset is the first of its kind in the domain of social media specifically tailored for the Assamese language. We are committed to making both the dataset and the models publicly accessible for the benefit of the research community. The dataset can be downloaded from Github[6].

2. Language Background and Related Work

Assamese, an eastern Indo-Aryan language, serves as the first language for nearly 15.3 million speakers (Chandramouli and General, 2011) and is recognized as one of the 22 scheduled languages in India. Its orthography, rooted in the Indic writing system, comprises 41 consonant and 11 vowel graphemes, employed to represent 8 vowel and 23 consonant phoneme sounds in Assamese (Mahanta, 2012). Unlike English, which utilizes 26 alphabets (5 vowels and 21 consonants) to produce 44 phoneme sounds (Bizzocchi, 2017), Assamese employs 52 alphabets for 31 phoneme sounds. This disparity leads to variations in the transliteration of Assamese words into Roman orthography when standard transliteration rules are not followed.

Recent years have seen notable advancements in Indic language transliteration. (Kunchukuttan et al., 2015) proposed *Brahmi-Net*, an online statistical transliteration system for transliteration and script conversion for all major Indian language pairs (18 languages and 306 language pairs) including Assamese. (Roark et al., 2020) introduced the Dakshina dataset, covering 12 South Asian languages in the Roman script, laying the foundation for transliteration and language modeling tasks. Extending this, (Kunchukuttan et al., 2021a) explored neural machine transliteration for English and 10 Indian languages, emphasizing multilingual transliteration. (Madhani et al., 2022) presented the Aksharantar dataset, the largest transliteration dataset covering 21 Indian lan-

[3] Sequitur

[4] Moses

[5] Google Input Tools

[6] https://github.com/osintg-iitghy/LREC-COLING-2024-OSINTG-IITG

guages, achieving state-of-the-art results with the IndicXlit model. In a recent study by (Ruder et al., 2023), an evaluation of sentence-level transliteration was conducted across 13 languages, encompassing 12 languages from the Dakshina dataset as well as the Amharic language, across 30 distinct transliteration directions. To carry out their experiments, the researchers harnessed transfer learning setups (mT5-Base, ByT5-Base, Flan-PaLM-62B). For social media like infomral text, the Forum for Information Retrieval (FIRE) organized pivotal shared tasks like FIRE 2013 and FIRE 2014 (Roy et al., 2013; Choudhury et al., 2014), focusing on Hindi song lyrics in Roman script. Despite these achievements, current research does not specifically address transliteration challenges in Romanized social media datasets.

3. Dataset Description

Our dataset was curated from three popular social media platforms, YouTube, Twitter (currently X), and Facebook using three publicly available APIs for systematic data extraction. Specifically, we employed the YouTube Data API[7] to harvest comments from predefined Assamese YouTube channels[8]. On Twitter (currently X), our focus was on acquiring reply tweets only from a prominent Assamese Twitter handle[9], utilizing the Tweepy API[10]. Similarly, we extracted comments from selected Assamese Facebook pages[11] using the Facebook Graph API[12]. Comprehensive details about the experimental dataset and the duration of data collection are available in Table 1.

In our dataset, we encompass a total of 60,312 sentences, ranging from single-word sentences to those extending up to 162 words. The average sentence length is 11.14, exhibiting a standard deviation of 8.38. Moreover, we note an average code-mixing percentage of 20.1% within the dataset. Code-mixing entails incorporating authentic Roman words alongside Romanized Assamese words, quantified as a percentage of the total words present in a sentence. At the word level, our dataset comprises a total of 671,921 words. Among them, 67,131 words are in English, 589,289 are Assamese words, and 15,501 are mixed-script words denoting a single token expressed in multiple scripts.

Out of the total 589,289 Assamese words, there are only 79,200 unique Assamese words, and from this set, we extracted a total of 65,614 unique transliteration pairs for conducting our experiments. Again, as the nature of social media data, a single source token can be represented in multiple ways. i.e., a single token can exhibit multiple transliteration variations. We have noticed a maximum of 127 Roman transliteration variations for a native Assamese word in our dataset. We have also noticed that based on the context or the similarity in pronunciation, a single Roman word may represent multiple native Assamese words, with one Roman word in our dataset representing a maximum of 31 native Assamese words. Figure 1 visually demonstrates the connection between the number of Roman variations and the corresponding count of native Assamese words exhibiting those many variations. Again, the plot in Figure 2 reveals the relationship between the number of Roman words and the total count of back-transliterated native Assamese words represented by those Roman words. It's worth noting that although not many terms in our dataset have the maximum number of variations, many words display more than one variation. Two examples in Table 2 shows these variations found in our dataset.

Table 1: Statistics of the collected dataset from three major social media sources along with the duration of data collection

Social Media Sources	Duration of Data Collection	#posts collected	#posts annotated	#words Assamese (total)	#unique Assamese words
Facebook	Dec-2013 to Feb-2017	409,168	5,300	71,800	79,200
YouTube	Jun-2018 to Aug-2023	385,676	50,000	426,089	
Twitter	Mar-2021 to Aug-2021	285,676	5,012	91,400	

4. Dataset Annotation

After acquiring the necessary dataset from our selected sources, we engaged 24 annotators and 3 linguistic experts as validators to annotate and verify the dataset. An online annotation tool, developed and deployed on our local server, facilitated this process. Annotators were selected based on their proficiency in both English and Assamese. A comprehensive annotation guideline was prepared in collaboration with linguistic experts. Both annotators and validators were required to register and log in to our system first. Upon logging in,

[7] YouTube Data API
[8] Dimpu's Vlogs, News Live, Assamese Mixture
[9] @himantabiswa
[10] Tweepy
[11] GU Confession Page, CMO Assam Page
[12] Facebook Graph API

Table 2: Two examples of both Roman and native variations along with the frequencies present in our dataset

Term	Script Language	Underlying Language	English Meaning	Total Variations	Number of Variations with Frequencies
"বহুত"	Assamese	Assamese	Many	27	***bohut:*** *2915,* ***bhut:*** *1047,* ***bht:*** *851,* ***bahut:*** *651,* ***bohot:*** *126,* ***buhut:*** *105,* ***bhout:*** *47,* ***bhot:*** *46,* ***bahot:*** *24,* ***bhtt:*** *13,* ***boht:*** *12,* ***bohud:*** *12,* ***bhoot:*** *8,* ***bout:*** *7,* ***buhot:*** *7,* ***bhohut:*** *6,* ***bohout:*** *6,* ***bhaut:*** *6,* ***vohut:*** *6,* ***bohoot:*** *5,* ***bohuuuuuuuuttttt:*** *1,* ***bohuuuuuut:*** *1,* ***vohot:*** *1,* ***bohuuuuuuuut:*** *1,* ***bohuuuuut:*** *1,* ***bohuuuut:*** *1,* ***bhhhuut:*** *1*
"gai"	Roman	Assamese	To Sing	5	গাই: *65,* গায়: *18,* গৈ: *3,* যায়: *1,* গান: *1*

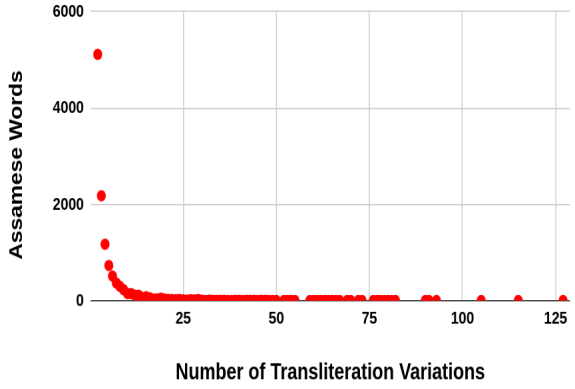

Figure 1: Distributions of Roman Transliteration Variations for Native Assamese Words.

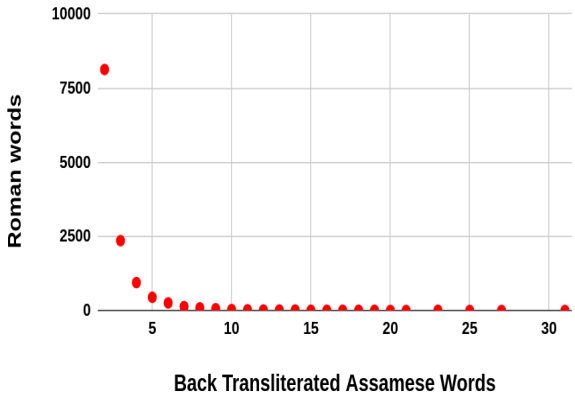

Figure 2: Distributions of Back Transliterated Native Assamese words represented by the Roman Words.

each annotator received an initial set of 100 sample posts for tagging, presented one at a time after submission. During tagging, annotators adhered to the annotation guidelines. Linguistic experts validated annotations using two tags: accept and reject. Those exceeding the 80% acceptance threshold in the initial test proceeded to the final annotation task. Annotators received a compensation of 3 INR for each accepted post.

The annotators were tasked with three main responsibilities. Firstly, they had to identify the language of the post at the sentence level, categorizing it as English, Assamese, Assamese-mixed, or Other. Secondly, identified Assamese words written in the Roman script were transliterated back to the corresponding Assamese words in the native script. English-origin words were retained if spelled correctly, otherwise replaced with accurate spelling. Thirdly, if a term in the sentence was recognized as a person's name, a geographical location, or the name of an organization, annotators selected and tagged the term or phrase as <person>, <place>, or <organization>, respectively, by clicking the appropriate label below the post.

Validators were responsible for reviewing and validating each post, marking it as accept or reject. In the case of rejection, validators provided explicit reasons for the rejection. Which will further reflected in the respective accounts of the annotators so that it can be tagged correctly in the subsequent attempts. The identical set of posts was allocated to two annotators to assess inter-annotator agreement in the reverse transliteration task at a later stage. We quantified inter-annotator agreement using Cohen's kappa (κ) coefficient and observed a value of 0.83. Figure 3 presents a snapshot of the annotation tool[13].

5. Experimental Setup

Our word-level transliteration experiments were conducted across four distinct setups: (1) Statistical Transliteration Setup, (2) Neural Network-Based Transliteration Setup, (3) Evaluation uti-

[13] https://www.iitg.ac.in/cseweb/osint/annotation/

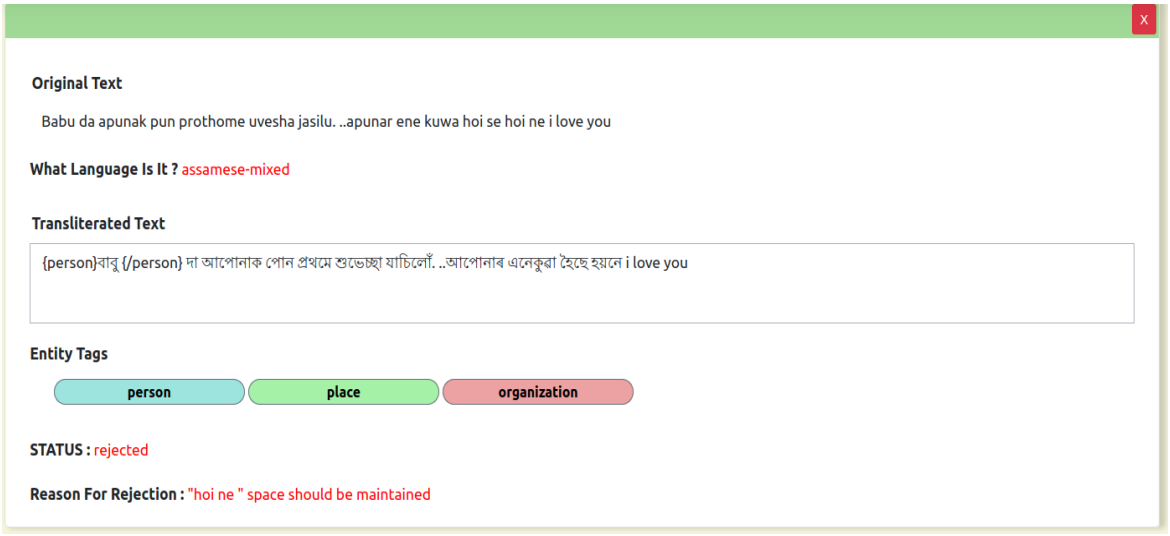

Figure 3: A snapshot of the annotation tool

lizing Pre-trained Transliteration Models, and (4). Transfer learning setup by fine-tuning existing state-of-the-art large language models (LLMs) with our word-level transliteration objective. In the subsequent discussion, each of these experimental setups will be briefly outlined.

1. **Statistical Machine Transliteration Setup:** In our statistical machine transliteration configurations, we employed two state-of-the-art statistical models for transliteration. One incorporated the phrase-based statistical machine transliteration model with the Moses decoder, employing GIZA++ (Och and Ney, 2003) for character alignment and KenLM (Heafield, 2011) for language modeling. The other model employed the joint n-gram-based string transduction system, SEQUITUR (Bisani and Ney, 2008). Both systems were trained up to 4-grams for the language model, and their respective performances were reported.

2. **Neural Network Based Transliteration Setup:** We mainly employed two state-of-the-art neural network based sequence-to-sequence models. One was implemented using the sequence-to-sequence BiLSTM encoder-decoder model with attention with the help of the OpenNMT toolkit (Klein et al., 2017) and the other using the Fairseq (Ott et al., 2019) implementation of neural transformer model.

    - **Experimental Setup for the BiLSTM with Attention Model:** For the BiLSTM model, the encoder is made up of two BiLSTM layers, each with 512 hidden units. The whole network processes words (split into characters) in batches of 128 embeddings, 64 training batches, and 32 validation batches, with an initial learning rate of 1 and a decay rate of 0.5 over 10,000 steps out of 200,000. BiLSTM dropout (Srivastava et al., 2014) and attention dropout with early stopping criterion with values of 0.3, 0.1, and 10 were used to reduce overfitting. Similarly, the decoder network includes 512 RNN (Sutskever et al., 2014) hidden units with 2 stacked on the decoder side. We use the Adagrad Optimizer (Duchi et al., 2011) to train the sequence-to-sequence system and normalize the gradient to prevent memory growth during training. At every 10,000 steps, we set a checkpoint.

    - **Experimental Setup for the Neural Transformer Model:** The Transformer model architecture consists of 6 encoder and 6 decoder layers, incorporating layernorm within each transformer layer for output normalization. The GELU activation function (Hendrycks and Gimpel, 2016) is applied, and both encoder and decoder self-attention layers utilize 4 attention heads. Parameters include a batch size of 1024, 256 dimensions for encoder and decoder embeddings, and 1024 dimensions for encoder and decoder feed-forward networks (FFN). Dropout rates of 0.5 and 0.1 are used, and label-smoothed cross-entropy with a smoothing factor of 0.1 serves as the training criterion. The Adam optimizer (Kingma and Ba, 2014) is employed with betas (0.9, 0.98) and a learning rate of 0.001. The learning rate scheduler is in-

verse square root with a warmup initialization learning rate of 0 and warmup updates of 4000. The model is trained for a maximum of 100 epochs.

3. **Pre-trained transliteration Model Evaluation:** We evaluated the transliteration performance of three publicly available pre-trained transliteration models directly on our test set, including the Google Transliteration API, the indic-trans transliteration model, and the multilingual transliteration model, IndicXlit.

4. **Transfer Learning Setup:** Additionally, we fine-tuned three models: the multilingual transliteration model IndicXlit, and two transformer-based large language models, the multilingual pre-trained text-to-text transformer model, mT5 and ByT5, a tokenizer-free variant of mT5, representing a token-free byte-to-byte pre-trained transformer model. Both pre-trained models were trained on the multilingual variant of the C4 dataset (Raffel et al., 2020), mC4 (Xue et al., 2021b) covering 101 languages. The pre-trained objective of the IndicXlit model aligns with ours, focusing on **"Word Level Transliteration"** only, and our target language, Assamese, is also present in the training set. In contrast, the pre-training objective for the other two models is **"Span Corruption"**, and our target language, Assamese, is not part of the training set in either of these two pre-trained models. For our evaluation, we utilized the mT5-small[14] and ByT5-small[15] pre-trained transformer models with 300 million parameters from the Hugging Face library, running both models for 51 epochs each.

6. Experimental Dataset

In our word-level transliteration task, we maintained uniformity in the training, validation, and test sets across all ten different setups. To ensure a thorough evaluation, we employed a 70-10-20 split. Out of the total 65,614 unique transliteration pairs, the training data includes 45,934 pairs, the validation set contains 6,560 pairs, and we evaluated the models with 13,120 pairs in the testing set.

7. Result and Discussion

In this section, we assess the performance of various transliteration setups using specialized evaluation metrics designed for word-level transliteration tasks. Given the nature of word-level transliteration, metrics such as Word Error Rate (WER), Character Error Rate (CER), the counts of substitution, deletion, and insertion errors, along with BLEU score (Papineni et al., 2002) (up to 4 grams), provide insightful measurements. WER represents the percentage of correctly predicted word pairs out of the total word pairs. CER, based on Levenshtein distance (Levenshtein, 1965), measures the minimum edits (substitution, insertion, and deletion) needed to transform a predicted word into the actual ground truth word. The BLEU score combines the Brevity Penalty and the Geometric Average of n-gram precision scores. We calculated BLEU scores up to 4 grams for each setup in this paper. Among the ten selected setups, those involving neural network-based models outperformed other baselines. Specifically, the setup with the neural transformer model (setup 4) achieved the lowest Word Error Rate (WER) and Character Error Rate (CER) values, along with the highest BLEU (up to 4 gram) score of 55.05, 19.44, and 69.15, respectively. The result of all the ten experimental setups are presented in Table 3.

(1). Statistical Model Setup: Within the statistical model setups, the Sequitur implementation of the joint n-gram-based transliteration model (setup 2) exhibited superior performance compared to its counterpart, the Phrase-based statistical transliteration model using Moses (setup 1), as indicated in the Table 3.

(2). Neural Model Setup: In the configurations featuring neural network-based transliteration models, the one employing the neural transformer model (setup 4) outperformed the BiLSTM model with attention (setup 3) in terms of Word Error Rate (WER), Character Error Rate (CER), and the BLEU (up to 4 gram) accuracy score, as depicted in Table 3.

(3). Pre-trained Model Evaluation Setup: Among the pre-trained models, the configuration utilizing the Google Transliteration API (setup 6) achieves comparable results with the best-performing neural transformer model (setup 4). It also demonstrates the highest performance among all three pre-trained model setups. However, the multilingual pre-trained transliteration model IndicXlit (setup 5), primarily trained on canonical transliteration pairs, encounters challenges with noisy transliteration pairs from social media. It is worth noting that most pre-trained transliteration models were trained on carefully annotated clean/canonical transliteration pairs, mapping from the native Assamese word to its respective romanized Assamese counterpart in the forward transliteration direction. In contrast, the pre-trained transliteration model indic-trans (setup 7)

[14] mt5-small
[15] byt5-small

Table 3: Transliteration result in terms of Word-Error-Rate(WER), Character-Error-Rate(CER), Number of Substitution, Insertion, Deletion errors and the BLEU (up to 4 gram) score for all the 10 setups (Experimental setups with the lowest word-error-rate(WER), lowest character-error-rate(CER) and the highest BLEU (up to 4 gram) score are highlighted in **bold**)

	Statistical Model Setup		Neural Model Setup		Pre-trained Model Evaluation Setup			Transfer Learning Setup		
	Phrase-based Statistical Transliteration model using Moses	Joint Source Channel based model using Sequitur	BiLSTM model with Attention	Neural Transformer model	IndicXlit model	Goggle Transliteration API	indic-trans model	IndicXlit model	Google's mT5 model	Google's ByT5 model
	setup 1	*setup 2*	*setup 3*	*setup 4*	*setup 5*	*setup 6*	*setup 7*	*setup 8*	*setup 9*	*setup 10*
WORD ERROR RATE (WER)	66.78	63.99	58.90	**55.05**	73.38	58.79	95.11	63.53	76.38	66.36
CHARACTER ERROR RATE (CER)	23.04	21.34	19.76	**19.44**	29.91	24.01	54.62	21.10	31.69	22.94
SUBSTITUTION ERROR	9146	8593	9728	8129	12265	9625	24507	8924	12633	8996
INSERTION ERROR	3093	3038	3311	2889	2946	3289	4021	2868	4340	3085
DELETION ERROR	3825	3258	3911	3289	5639	3821	9550	2921	5119	3911
BLEU (up to 4 gram) SCORE	64.41	67.33	68.60	**69.15**	54.03	67.61	24.04	66.48	55.50	65.93

exhibits the lowest performance among all ten experimental setups, as indicated in Table 3.

(4). Transfer Learning Setup: In the transfer learning setup, we fine-tuned three pre-trained models. One utilized the transformer-based multilingual pre-trained transliteration model, IndicXlit, trained on 22 Indic languages, including our target language Assamese in their training set. Additionally, we employed two multilingual transformer-based pre-trained large language models trained on 101 languages from the mC4 dataset, where Assamese is not part of the training set. Furthermore, their pre-training objective, **"Span Corruption"** differs from our "**word-level transliteration**" objective. In contrast to the character-level embedding used in earlier setups, the mT5 model (setup 9) employs pre-trained SentencePiece embedding to create the vocabulary. On the other hand, the ByT5 model (setup 10) follows a language-agnostic approach, directly processing UTF-8 bytes without any text pre-processing. These bytes are embedded into the model's hidden size using a vocabulary of 256 possible byte values. Since ByT5 operates on the byte level rather than the character or sub-word level, it doesn't maintain a fixed vocabulary size, enabling it to process text in any language, given that characters in any language have unique UTF-8 byte values. Among the three transfer learning setups, the fine-tuned IndicXlit model achieves the highest performance in terms of Word Error Rate (WER), Character Error Rate (CER), and BLEU (up to 4 gram) accuracy score, as depicted in Table 3.

8. Challenges in Social Media Transliteration

Processing social media text poses several challenges, including generic transliteration issues and those arising from the inherent nature of noisy social media content. The challenges are amplified by users not adhering to standard transliteration guidelines while writing Assamese using the Roman script on social media. Additionally, English and Assamese are orthographically distinct languages, lacking a direct one-to-one correspondence between their graphemes. Many social media users are comfortable with the English QWERTY keyboard, leading them to phonetically transcribe Assamese using Roman alphabets based on their own phonetic judgment. Transliteration models often struggle to address these challenges, leading to errors. In the subsequent discussion, we will discuss some of those challenges encountered while processing social media text. The observations are categorized into four different categories, outlined in Table 4, alongside some sample examples. Model outputs that align with the ground truths are highlighted in blue.

(1). Multiple character mapping: Due to the absence of direct one-to-one correspondence between English and Assamese graphemes, coupled with the lack of adherence to a common standard transliteration guideline, we have observed instances where a single Roman grapheme represents multiple Assamese graphemes (one-to-many mapping). Conversely, a single Assamese grapheme may exhibit various Roman variations

Table 4: Comparison between the outputs of ten different transliteration model setups with the same Roman input (output of the setups that match with the ground truths are highlighted in blue)

Different Setups	Roman Input and Actual Native ground truth		Statistical Model Setups		Neural Model Setups		Pre-trained Model Evaluation Setups			Transfer Learning Setups		
Different Obsevations	Roman Input	Assamese Native	*Moses output (setup1)*	*Sequitur output (setup2)*	*BiLSTM output (setup3)*	*Transformer output (setup4)*	*IndicXlit model output (setup5)*	*Google Transliteration API output (setup6)*	*indic-trans model output (setup7)*	*Fine-tune IndicXlit model output (setup8)*	*Fine-tune mT5 model output (setup9)*	*Fine-tune ByT5 model output (setup10)*
Multiple Character Mapping	hani	শনি	হানি	হানি	শনি	সানি	হানি	সানি	হানী	হানি	সানি	হানি
	xuola	শুৱলা	শুলা	সোলা	খুৱলা	খোৱালা	সোঁৱলা	শুৱলা	জুওলা	শুৱলা	সোলা	শুৱলা
Short form Representation	jrhtt	যোৱহাটত	যোৱহাতত	যোৱহাতত	যোৱহাতত	যোৱহাটত	জাৰ্ট	হৰ্দ	জৰত্ত	যোৱহাট	যোৱহাতত	যোৱহাতত
	bhtor	বহুতৰ	বহঁতৰ	বহঁতৰ	বহঁতৰ	বহুতৰ	ভটৰ	ভাতৰ	ভ্টর	ভাতৰ	ভিতৰ	ভটৰ
Long form Representation	aaauuu	আওঁ	আওঁ	আ	আওঁ	আওঁ	আওঁওও	আআওওও	আউ	আও	আও	আআও
	bapppaaoiii	বাপ্পাঐ	বাপপায়	বাপ্পাও	বাপপ্পাওঁ	বাপপ্পাওঁ	বাপ্পাআওই	বাপ্পাপায়ী	বাপ্পাওঈ	বাপ্পায়	বাপ্পাঐ	বাপ্পাঐ
Alphanumeric Word	ai2e	এইটোৱে	এইটো	এইটো	এইটোৱে	এইটোৱে	এআইআইচে	অং২য়ে	আই2এ	এইটোৱে	এইটোৱে	এইটোএ
	kn2	কিন্তু	কিন্তু	কোনতো	কিন্তু	কিন্তু	কেএনচি	কঁ২	ন2	কিন্তু	কিন্তু	কিন্তু

(many-to-one mapping). In our dataset, for instance, the Roman grapheme "r" corresponds to Assamese graphemes "ৰ" "ড়", and "ঢ়" in the transliteration of "ৰং" (transliteration: "**r**ang", meaning: "Colour"), "গড়" (transliteration: "go**r**", meaning: "Rhino") and "বুঢ়া" (transliteration: "bu**r**a", meaning: "Old man"). Similarly, the Assamese grapheme "শ" is transliterated with different Roman graphemes such as "s", "sh", "h", "x", and "kh". For example, in the representation of the Assamese word "শিলাদিত্য", it is transliterated as "**s**iladitya", "**sh**iladitya", "**h**iladitya", "**x**iladitya", and "**kh**iladitto" as evidenced by our dataset. Referencing Table 4, we have observed two instances of Assamese words, "শনি" and "শুৱলা", where the Assamese grapheme "শ" is correspondingly represented as "h" and "x" in the ground truth. Consequently, due to these multiple character mappings, our models predicted "শ" as "হ", "স", "খ" and "জ" respectively.

(2). Short form representation: In social media, it is a common practice to utilize informal short-form representations by excluding vowels between consonants in Roman transliterations. When individuals write Assamese words using Roman characters on social media, they intentionally omit vowels between consonants to create short-form versions. For instance, the Assamese word "লগত" is transliterated as **"lgt"** in a short form by removing the vowels between consonants. However, the ideal Roman transliteration of the Assamese word "লগত" should be either **"lagat"** or **"logot"** without the short form. Table 4 presents two illustrative examples of short-form representations derived from our datasets: **"jrhtt"** ("যোৱহাটত") and **"bhtor"** ("বহুতৰ"). Notably, only the Transformer model in setup 4 yielded outputs that accurately matched the ground truth in both cases.

(3). Long form representation: In social media text, repeating the same character multiple times to emphasize or convey emotions is common. For example, the Assamese word "বহুত" might be transliterated as **"bohuuuuuuuuttttt"** instead of its standard form **"bohut"**. Furthermore, In the Assamese language, the presence or absence of the nasalized character "ঁ" holds significant meaning, conveying different interpretations. For instance, "কাহ" (meaning: cough) and "কাঁহ" (meaning: bell metal) showing this distinction. We offer two examples of long-form representation from our dataset: **"aaauuu"** ("আওঁ") and **"bapppaaoiii"** ("বাপ্পাঐ") in Table 4. The presence of the nasalized character "ঁ" is correctly predicted by the statistical model with Moses (setup 1) and both the neural models, namely the BiLSTM with attention model in setup 3 and the Transformer model in setup 4. Similarly, the long-form representation of "বাপ্পাঐ" for **"bapppaaoiii"** is accurately predicted by the transfer learning setups in setup 9 (mT5) and setup 10 (ByT5).

(4). Alphanumeric word: A common practice noted in the transliteration of Assamese social media text involves representing a single word by combining both alphabetic letters and numbers within the same word. For example, the Assamese words "এইটোৱে" and "কিন্তু" are transliterated as **"ai2e"** and **"kin2"** respectively, as demonstrated in Table 4 of our dataset. It is worth noting that, except for the pre-trained models in setup 5, setup 6, and setup 7, all the other models effectively accommodate this alphanumeric pattern and correctly predicted the output, as evident in our dataset.

9. Conclusion and Future Work

In summary, this paper has introduced a pioneering back transliteration dataset for Assamese, capturing diverse linguistic content from three popular social media platforms such as Facebook, Twitter (currently X), and YouTube. We also conducted a comprehensive evaluation of ten state-of-the-art word-level transliteration benchmarks. The experimental evaluation highlighted the superiority of the Neural Transformer model, achieving the lowest Word Error Rate (WER) and Character Error Rate (CER), along with the highest BLEU (up to 4 gram) score.

Looking forward, this domain has several scope for future research and development. Firstly, expanding the dataset to include more diverse linguistic content and covering additional social media platforms could enhance the robustness of transliteration models. Addressing specific challenges, such as variations in informal short-form and long-form representations, capturing instances of multiple character mappings, and identifying and handling other social media-specific challenges would contribute to more accurate transliterations. In terms of model development, fine-tuning existing pre-trained models on the unique characteristics of social media transliterations and exploring transformer-based architectures tailored for low-resource languages like Assamese could yield improvements.

This work establishes the groundwork for enhancing transliteration technology, particularly for low-resource languages and within the context of social media. Our focus is on promoting cross-lingual communication and facilitating resource development. Looking ahead, our efforts will extend to tackling sentence-level transliteration and addressing the code-mixed nature of social media text. Given the linguistic diversity prevalent in India and globally, overcoming these transliteration challenges holds significance for diverse social media scenarios and languages.

10. Acknowledgements

This dataset has been generated in Open Source Intelligence Lab, Indian Institute of Technology, Guwahati, and has been partially funded by the Ministry of Electronics & Information Technology, Government of India.

11. Bibliographical References

Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*.

Hemanta Baruah, Sanasam Ranbir Singh, and Priyankoo Sarmah. 2023. Assamese back transliteration-an empirical study over canonical and non-canonical datasets. In *Proceedings of the 37th Pacific Asia Conference on Language, Information and Computation*, pages 801–808.

Hemanta Baruah, Sanasam Ranbir Singh, and Priyankoo Sarmah. 2024. Transliteration characteristics in romanized assamese language social media text and machine transliteration. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 23(2).

Irshad Ahmad Bhat, Vandan Mujadia, Aniruddha Tammewar, Riyaz Ahmad Bhat, and Manish Shrivastava. 2014. Iiit-h system submission for fire2014 shared task on transliterated search. In *Proceedings of the 6th Annual Meeting of the Forum for Information Retrieval Evaluation*, FIRE '14, page 48–53, New York, NY, USA. Association for Computing Machinery.

Maximilian Bisani and Hermann Ney. 2008. Joint-sequence models for grapheme-to-phoneme conversion. *Speech communication*, 50(5):434–451.

Aldo Luiz Bizzocchi. 2017. How many phonemes does the english language have? *International Journal on Studies in English Language and Literature (IJSELL)*, 5(10):36–46.

C Chandramouli and Registrar General. 2011. Census of india. *Rural Urban Distribution of Population, Provisional Population Total. New Delhi: Office of the Registrar General and Census Commissioner, India*.

Monojit Choudhury, Gokul Chittaranjan, Parth Gupta, and Amitava Das. 2014. Overview of fire 2014 track on transliterated search. *Proceedings of FIRE*, pages 68–89.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7).

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Kenneth Heafield. 2011. Kenlm: Faster and smaller language model queries. In *Proceedings of the sixth workshop on statistical machine translation*, pages 187–197.

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. *arXiv preprint arXiv:1701.02810*.

Kevin Knight and Jonathan Graehl. 1997. Machine transliteration. *arXiv preprint cmp-lg/9704003*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Anoop Kunchukuttan, Siddharth Jain, and Rahul Kejriwal. 2021a. A large-scale evaluation of neural machine transliteration for Indic languages. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3469–3475, Online. Association for Computational Linguistics.

Anoop Kunchukuttan, Siddharth Jain, and Rahul Kejriwal. 2021b. A large-scale evaluation of neural machine transliteration for indic languages. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3469–3475.

Anoop Kunchukuttan, Ratish Puduppully, and Pushpak Bhattacharyya. 2015. Brahmi-net: A transliteration and script conversion system for languages of the Indian subcontinent. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 81–85, Denver, Colorado. Association for Computational Linguistics.

Vladimir I. Levenshtein. 1965. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics. Doklady*, 10:707–710.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Yash Madhani, Sushane Parthan, Priyanka Bedekar, Ruchi Khapra, Vivek Seshadri, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh M. Khapra. 2022. Aksharantar: Towards building open transliteration tools for the next billion users.

Shakuntala Mahanta. 2012. Assamese. *Journal of the International Phonetic Association*, 42(2):217–224.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51.

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, Minneapolis, Minnesota. 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.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Brian Roark, Lawrence Wolf-Sonkin, Christo Kirov, Sabrina J. Mielke, Cibu Johny, Isin Demirsahin, and Keith Hall. 2020. Processing South Asian languages written in the Latin script: the Dakshina dataset. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2413–2423, Marseille, France. European Language Resources Association.

Rishiraj Saha Roy, Monojit Choudhury, Prasenjit Majumder, and Komal Agarwal. 2013. Overview of the fire 2013 track on transliterated search. In *Proceedings of the 4th and 5th Annual Meetings of the Forum for Information Retrieval Evaluation*, FIRE '12 & '13, New York, NY, USA. Association for Computing Machinery.

Sebastian Ruder, Jonathan H Clark, Alexander Gutkin, Mihir Kale, Min Ma, Massimo Nicosia,

Shruti Rijhwani, Parker Riley, Jean-Michel A Sarr, Xinyi Wang, et al. 2023. Xtreme-up: A user-centric scarce-data benchmark for under-represented languages. *arXiv preprint arXiv:2305.11938*.

Snigdha Singhania, Minh Nguyen, Gia H Ngo, and Nancy Chen. 2018. Statistical machine transliteration baselines for news 2018. In *Proceedings of the Seventh Named Entities Workshop*, pages 74–78.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2021a. Byt5: Towards a token-free future with pre-trained byte-to-byte models.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021b. 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.