When Scripts Diverge: Strengthening Low-Resource Neural Machine Translation Through Phonetic Cross-Lingual Transfer

Ammon Shurtz, Christian Richardson, Stephen D. Richardson
Brigham Young University, USA
{acshurtz, richachr, srichardson}@byu.edu

Abstract

Multilingual Neural Machine Translation (MNMT) models enhance translation quality for low-resource languages by exploiting cross-lingual similarities during training—a process known as knowledge transfer. This transfer is particularly effective between languages that share lexical or structural features, often enabled by a common orthography. However, languages with strong phonetic and lexical similarities but distinct writing systems experience limited benefits, as the absence of a shared orthography hinders knowledge transfer. To address this limitation, we propose an approach based on phonetic information that enhances token-level alignment across scripts by leveraging transliterations. We systematically evaluate several phonetic transcription techniques and strategies for incorporating phonetic information into NMT models. Our results show that using a shared encoder to process orthographic and phonetic inputs separately consistently yields the best performance for Khmer, Thai, and Lao in both directions with English, and that our custom Cognate-Aware Transliteration (CAT) method consistently improves translation quality over the baseline.

1 Introduction

A common approach to enhancing Neural Machine Translation (NMT) for low-resource languages involves leveraging the knowledge from similar high-resource languages. One approach to this is **multilingual learning**, in which a high-resource language pair is combined with a low-resource language pair within a single multilingual model (Chen et al., 2019)

This method is effective with large models that support dozens of languages (Aharoni et al., 2019; Gala et al., 2023). The method also performs well on a smaller scale when pairing related languages, such as low-resource Haitian Creole with high-resource French (Robinson et al., 2023), Vietnamese and French (Ngo et al., 2020), or Catalan and several higher-resource Indo-European languages (Chen and Abdul-Mageed, 2021). In these cases, the low-resource language improvements are enabled by the token overlap with the higher-resource languages (Aji et al., 2020; Patil et al., 2022). This token overlap relies on the shared scripts between the high- and low-resource languages, a benefit not all low-resource languages have (Muller et al., 2021).

Some low-resource languages have a related high-resource counterpart but use a different writing system. Despite strong phonetic and lexical similarities, the lack of a shared writing system almost completely eliminates token overlap, potentially limiting the benefits of transfer learning. One way to address this problem is by increasing token overlap; for example, Limisiewicz et al. (2023) achieve this by modifying the tokenizer, though our approach differs.

In this work, we propose and evaluate a method for increasing token overlap in NMT models through the use of phonetic transliterations. Specifically, we incorporate both phonetic information and the original orthographic representations of three Southeast Asian languages into a Multilingual NMT (MNMT) model. Our evaluation focuses on Thai, Lao, and Khmer—closely related languages spoken in Thailand, Laos, and Cambodia, respectively. Although these languages share many lexical and grammatical similarities, each employs a distinct orthographic system.

We compare a baseline multilingual NMT (MNMT) system, which uses only the orthographic representations of the languages, against three transliteration methods. The transliteration methods include International Phonetic Alphabet (IPA) transcriptions, Romanization, and a custom method we call Cognate-Aware Transliteration (CAT). These transcriptions are integrated

Proceedings of the 5th Workshop on Multilingual Representation Learning (MRL 2025), pages 336–346

with the original orthographies in three ways: 1) by concatenating the orthographic and transliterated representations as a single input to a vanilla transformer, 2) by using a single encoder that processes the two inputs separately before concatenating their embeddings for a shared decoder, and 3) by using two separate encoders—one for the orthographic input and one for the transliterated input—combined with a shared decoder. More details can be found in Section 3.

Incorporating phonetic information allows MNMT models to overcome divergent orthographies and improve knowledge transfer between languages, boosting translation quality by up to 3.4 BLEU points and 4.4 chrF points for low-resource Southeast Asian languages. Additional results show that IPA and CAT generally outperform Romanization, with shared-encoder models achieving the largest gains over the baseline. Overall, we contribute:

- A framework for integrating phonetic transliterations into multilingual NMT.
- Cognate-Aware Transliteration (CAT), a novel method for capturing cross-lingual similarities.
- A comprehensive evaluation of transliteration and integration strategies on Thai, Lao, and Khmer.

2 Related Works

Previous research has been conducted for the cross-lingual transfer of various NLP tasks in Chinese, Japanese, Korean, and Vietnamese (CJKV). Nguyen et al. (2023) utilize the International Phonetic Alphabet (IPA) to produce transcriptions in an attempt to improve the cross-lingual transfer for CJKV languages. They show improvements in cross-lingual transfer for POS tagging and NER tasks. Nguyen et al. (2024) build on that work by creating more benchmark data for additional tasks beyond token-level POS tagging and NER. Romanization is also included in experiments in addition to the phonetic transcriptions, finding the romanization to perform better than the phonetic transcriptions. Both of these works focus on the alignment of the transcriptions/romanization to the orthographic tokens. Moosa et al. (2023) further study transliteration as a cross-lingual signal for Indic languages, showing that transliteration can improve multilingual language modeling and downstream task performance across scripts.

Recent work extends these ideas to large language models (LLMs). Purkayastha et al. (2023) propose a large-scale romanization-based adaptation approach for multilingual LLMs, demonstrating improved transfer to low-resource and non-Latin languages. Similarly, J et al. (2024) introduce RomanSetu, which leverages romanization to improve multilingual capabilities in LLMs while reducing training costs. Nguyen et al. (2025) explore phoneme-based prompting for LLMs, finding that phonemic representations enhance multilinguality for non-Latin-script languages.

Romanization has been used to enhance knowledge transfer in multilingual NMT models. A universal parent model trained with a Romanized vocabulary was found to achieve improved knowledge transfer in a many-to-one translation scenario (Gheini and May, 2019). Amrhein and Sennrich (2020) extended this approach to many-to-many NMT models and found that while romanization does not consistently improve results across all languages, it is beneficial in cases where related languages use different scripts. In such scenarios, romanization facilitates knowledge transfer. Additionally, Salesky et al. (2023) address this problem by abstracting vocabularies entirely. They utilize multilingual pixel representations, enabling the model to generalize to new and even unseen scripts as inputs.

While prior work has applied romanization and phonetic representations to well-resourced language families, our study focuses on lower-resource Southeast Asian languages with limited transliteration tools in the underexplored domain of Neural Machine Translation.

3 Methodology

In Section 3.1, we describe the non-transliterated baseline inputs and the three transliteration methods we intend to compare. In section 3.2 we describe the methods for computing token overlap between transliterated texts. Finally, section 3.3 describes the methods for integrating the phonetic transcriptions into NMT models.

3.1 Phonetic Transcriptions

There are multiple levels of granularity at which phonetic transcriptions can be applied. In this work, we explore whether different translitera-

tion strategies affect downstream model performance. By varying the degree of token overlap across languages—from none at all to a highly customized scheme designed to maximize overlap—we aim to understand how transcription choices influence cross-lingual modeling. The following subsections describe the four approaches we evaluate, ranging from no transliteration to a cognate-aware system.

No transliteration. As a baseline, we evaluate the models without any transliteration, using the original orthographic representations of the text for all languages. We expect this to have the lowest amount of token overlap between related languages of different scripts.

International Phonetic Alphabet (IPA). We consider the most granular method for transliteration to be converting text into IPA transcriptions. IPA would maintain the most subtle differences between languages and dialects, which could be detrimental to this methodology. Despite this, we expect that the unified alphabet will still yield much more token overlap than original orthographies.

Romanization. Romanization is the process of converting text from another script into the Latin alphabet. We expect that transliterating non-Latin scripts into Latin would be result in simpler transcriptions compared to IPA, but still be granular enough to be useful in distinguishing sounds.

Cognate-Aware Transliteration (CAT). *Regular sound correspondences* are systematic phoneme changes that occur in cognates across related languages (Brown et al., 2013). For example, the /tɕʰ/ sound in Thai is systematically replaced by the /s/ sound in Lao. Similarly, the Thai /r/ is replaced by /h/ or /l/ in Lao. Additionally, similar substitutions occur between some German and English words, such as the replacement of the English /ð/ sound in "this" and "that" with the /d/ sound in their German equivalents, "dies" and "das."

For this method, sound correspondences would be represented by unified characters for both languages in the transliteration, with the purpose of representing cognates uniformly. There are currently no automatic methods for finding regular sound correspondences and thus CAT rules would need to be created manually for a set of languages, though one potential method could be to automatically detect cognates based on parallel data (Grönroos et al., 2018) and then use those to create a CAT system. We hypothesize that a high quality transliteration system based on the regular sound correspondences between languages would yield the highest overlap of tokens, compared to the previous methods.

3.2 Vocabulary Overlap

In multilingual NLP models, shared vocabularies between languages are commonly used. Previous work has shown that larger vocabulary overlap leads to improved model performance (Pires et al., 2019; Wu and Dredze, 2019). Our work seeks to determine whether this applies to Neural Machine Translation, and more specifically if the amount of vocabulary overlap between the transliterations (not the original orthographies) correlates with downstream translation performance.

To assess the degree of vocabulary overlap between languages, we employ two metrics. These metrics are based on discrete token-level overlap comparisons using the Jaccard Index (Jaccard, 1901), defined as:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{1}$$

Corpus-level Jaccard (CJ). This is the simplest metric for quantifying vocabulary overlap. We compute the Jaccard Index at the corpus level, where set A contains all unique tokens in Language A, and set B contains all unique tokens in Language B. This metric provides a general sense of phonetic overlap between the two languages based on their transliterations. However, it does not capture whether semantically equivalent sentences share a high degree of lexical overlap.

Mean Pairwise Jaccard (MPJ). We define Mean Pairwise Jaccard (MPJ) as the average Jaccard Index computed between aligned sentence pairs across two languages. For each sentence pair i, let A_i denote the set of unique tokens in sentence i in language A, and B_i denote the corresponding set of unique tokens in the translated sentence in language B.

We define two vectors of sets:

$$\mathbf{a} = (A_1, \ldots, A_n), \quad \mathbf{b} = (B_1, \ldots, B_n)$$

MPJ is then computed as:

$$\text{MPJ}(\mathbf{a}, \mathbf{b}) = \frac{1}{n} \sum_{i=1}^{n} J(A_i, B_i) \tag{2}$$

where $J(A_i, B_i)$ is the Jaccard Index between the token sets of sentence i.

This metric better captures whether semantically equivalent sentences share a high degree of lexical overlap.

3.3 Phonetic Integration

Neural Machine Translation (NMT) models aim to generate a target sentence $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ given a source sentence $\mathbf{x} = (x_1, x_2, \ldots, x_m)$. The model defines a conditional probability distribution:

$$P(\mathbf{y} \mid \mathbf{x}) = \prod_{i=1}^{n} P(y_i \mid y_1, \ldots, y_{i-1}, \mathbf{x}) \quad (3)$$

Each term in the product represents the probability of generating the token y_i at step i given all previously generated tokens $y_1, \ldots, y_{i-1}$ and the entire source sentence $\mathbf{x}$.

To incorporate phonetic information, we introduce a transcription function $\tau(\mathbf{x})$ that maps the source sentence to its phonetic representation. The conditional probability is then modified to condition each target token not only on the previously generated tokens and the source sentence in its original script, but also the phonetic transcription:

$$P(\mathbf{y} \mid \mathbf{x}) = \prod_{t=1}^{n} P(y_i \mid y_1, \ldots, y_{i-1}, \mathbf{x}, \tau(\mathbf{x})) \quad (4)$$

The target output of the NMT model can be conditioned on a given transcription function $\tau(\mathbf{x})$ in various ways. We propose the following methods for integrating the phonetic transcriptions:

Concatenated Input. Orthographic and phonetic sequences are concatenated into a single input.

Shared Encoder. A single encoder processes both inputs; their embeddings are concatenated before decoding.

Dual Encoder. Separate encoders process orthographic and phonetic inputs, with a shared decoder attending to both.

4 Experiments

4.1 Data

To evaluate the various phonetic transcription and integration methods, we study the following set of South-East Asian languages: Khmer, Lao, and Thai. Although each language uses a distinct writing system, they share significant linguistic similarities because of common historical and geographical background, with roots in Pali and Sanskrit (Enfield, 2019).

Language Pair	Uncleaned	Cleaned
Thai - English	2,175,880	1,080,329
Lao - English	1,994,050	612,836
Khmer - English	1,501,301	501,955

Table 1: Approximate number of parallel segments for each language pair. Extensive cleaning was performed to ensure higher quality data.

We utilize the Paracrawl Bonus dataset which focuses on better coverage for South and East Asian languages (Koehn, 2024). This data is noisy, so we applied the guidelines found in the GILT Leaders Forum's Best Practices in Translation Memory Mangement.[1] Details of the cleaning pipeline are provided in Appendix A; the most impactful step was validating that Unicode characters correspond to the intended language. Table 1 shows the number of parallel sentence pairs for each language pair before and after cleaning.

We use all the cleaned data for training, in both English $\rightarrow$ X and X $\rightarrow$ English directions. For validation and testing, we use the FLORES+ (NLLB Team et al., 2024) dev and devtest datasets for each language direction, ensuring that there was no data contamination in the training set.

4.2 Transliteration

Although several IPA transliteration tools are available for Thai (Phatthiyaphaibun et al., 2023), and the Uroman package (Hermjakob et al., 2018) provides coverage for all three languages under study, we chose to develop our own transliteration software and typology for IPA, romanization, and CAT (Cognate-aware Transliteration), which we release on our public Github repository.[2] This ensured that our comparisons remained consistent and fair, avoiding the inconsistencies that can arise when relying on multiple tools created by different designers. Additionally, there is currently a distinct lack of quality, openly-available transliteration software for Khmer and Lao.

[1] https://github.com/GILT-Forum/TM-Mgmt-Best-Practices/blob/master/best-practices.md

[2] https://github.com/byu-matrix-lab/sea-transliteration-mnmt

Transliteration Method	Thai Sentence	Khmer Sentence	Token Overlap
None	คุณสามารถเรียนภาษาที่มหาวิทยาลัยได้	អ្នកអាចរៀនភាសានៅមហាវិទ្យាល័យបាន។	0
IPA	kʰunsaːmaːrtʰrianpʰaːsaː tʰiːmhaːuaitʰjaːlajajd	ʔnkʔaːcriənpʰaːsaːnaw mhaːʋetjaːljbaːn.	4
Romanization	khunsaamaarthrianphaasaa thiimhaauaithyaalaiaid	qnkqaacrienphaasaanau mhaavityaalybaan.	4
CAT	khonsaamaartrianphaasaa teimhaauaityaalaeaet	ɔnkɔaacrianphaasaanao mhaavityaalybaan.	5

Table 2: Example Thai and Khmer translations of the English sentence: "You can learn languages at a university." Each sentence is transliterated using the International Phonetic Alphabet (IPA), romanization, and a custom Cognate-Aware Transliteration (CAT). Each representation is tokenized using the XLM-RoBERTa (Conneau et al., 2020) tokenizer and the overlap of tokens between the two sequences is calculated as the intersection between the two token sets.

To unify our transliteration methods, we created a simple transliteration script that replaces specified Unicode characters with others based on a JSON file containing all mappings. This supports single Unicode characters and sequences of Unicode characters.

For our IPA transliterations, we used the Wikipedia script descriptions from the Khmer[3], Thai[4], and Lao[5] script pages. For romanization, we used the mappings described in the Uroman (Hermjakob et al., 2018) source code.[6]

To create a transliteration scheme that heavily encourages token overlap between languages, we created CAT for the three South-East Asian languages. This was designed by categorizing each consonant and vowel character in each of the languages according to both orthographic similarity and phonetic similarity. More details on the creation of CAT for Khmer, Lao, and Thai are contained in Appendix B.

To showcase the differences for each of these methods, we provide an example in Table 2. In this example, we take a Thai and a Khmer translation of the sentence "You can learn languages at a university." and transliterate using the four methods: None, IPA, Romanization, and CAT. These transliterations are tokenized using the XLM-RoBERTA (Conneau et al., 2020) tokenizer to demonstrate token overlap differences.

[3]https://en.wikipedia.org/wiki/Khmer_script
[4]https://en.wikipedia.org/wiki/Thai_script
[5]https://en.wikipedia.org/wiki/Lao_script
[6]https://github.com/isi-nlp/uroman

4.3 Training Implementation

For our experiments, we compare a baseline Transformer (Vaswani et al., 2017) model to each combination of transliteration and integration method, resulting in nine model variants. The transliteration methods are (1) IPA transcriptions, (2) Romanization, and (3) our proposed Cognate-Aware Transliteration (CAT). Each is integrated into the model using one of three approaches: (a) concatenating orthographic and transliterated inputs, (b) processing them separately within a shared encoder before concatenation at the embedding level, or (c) using two separate encoders combined with a shared decoder.

All experiments are based on the Transformer-base architecture. We use the *BARTForConditionalGeneration* implementation (Lewis et al., 2019), modified to support both the shared-encoder and dual-encoder configurations.

Each model contains 6 encoder layers and 6 decoder layers, with the dual-encoder setup allocating 6 layers to each encoder. The feed-forward network has a dimensionality of 2048, each encoder and decoder uses 8 attention heads, and the hidden size (d_{model}) is 512. We employ ReLU activations and apply dropout with a rate of 0.1.

Models are trained to convergence using 8 A100 GPUs, with an effective batch size of 8,192. Validation is performed every 4,000 steps, and convergence is determined using the validation set.

For tokenization, we train Byte-Level BPE tokenizers using the HuggingFace *Tokenizers* li-

		Orth.	IPA	Rom.	CAT
Tha–Lao	CJ	0.024	0.230†	0.198	**0.719**
	MPJ	0.029	0.113†	0.093	**0.394**
Tha–Khm	CJ	0.007	0.055	0.107†	**0.694**
	MPJ	0.011	0.062†	0.060	**0.202**
Khm–Lao	CJ	0.007	0.042	0.080†	**0.637**
	MPJ	0.011	0.065†	0.064	**0.198**

Table 3: Corpus-level Jaccard (CJ) and Mean Pairwise Jaccard (MPJ) scores for Thai (Tha), Lao (Lao), and Khmer (Khm) across four transliteration methods: native orthography (Orth.), IPA, Romanization (Rom.), and CAT. Bold = highest overlap; † = second highest.

brary.[7] We build separate multilingual tokenizers for each representation—orthography-only, IPA, Romanization, and CAT—each with a vocabulary size of 32K, trained on uniformly sampled sentences from the training set. For the shared-encoder and concatenation models, we train joint tokenizers that include both orthographic and transliterated text, using a larger vocabulary size of 56K, also drawn from uniformly sampled training data.

5 Results and Discussion

5.1 Vocabulary Overlap

To determine vocabulary overlap for each transliteration method, we first created the "complete" (Freitag and Firat, 2020) aligned data so we can compare sentences across non-english centric pairs, using English as a pivot to find the X → Y translation directions. This resulted in 19,525 sentences translated into Khmer, Lao, and Thai.

We calculated Corpus-level Jaccard (CJ) and Mean Pairwise Jaccard (MPJ) for the following language pairs across each transliteration method: Thai ↔ Lao, Thai ↔ Khmer, and Khmer ↔ Lao. Each language was transliterated into IPA, Romanization, and CAT and we report overlap metrics in Table 3, with the original orthography overlap calculations included as a baseline reference. Overlap is determined using the tokenizers trained for each transliteration method, as described in Section 4.3.

As expected, the overlap between tokens when using the native orthographies is close to 0, indicating almost zero overlap. The little overlap that is included is likely to be punctuation and numerals common to all three languages. Meanwhile, we see that CAT achieves the highest amount of overlap both globally and at the sentence-level. For the more linguistically related Thai–Lao pair, IPA yields greater token overlap than Romanization, whereas the Khmer–Thai and Khmer–Lao pairs show lower values and mixed outcomes between IPA and Romanization.

5.2 Multilingual Neural Machine Translation (MNMT)

We report chrF++ (Popović, 2017) and BLEU (Papineni et al., 2002) scores for all language directions calculated using SacreBLEU (Post, 2018). For language directions with English as the target, we utilize the default tokenization for BLEU. For language directions with English as the source, we utilize the Flores-200 tokenizer to calculate an spBLEU score instead, as the South-East Asian languages do not use spaces as word delimiters.

A summary of all chrF++ and BLEU/spBLEU scores are shown in Table 4. Overall, all transliteration methods and integration methods generally improve over the baseline, as indicated by a higher score with statistical significance. The gains appear to be larger when translating into English, reflecting the baseline's struggle to encode and comprehend the South-East Asian languages. Using a shared encoder with IPA transliterations achieves the highest scores in all but 1 direction, all of which are statistically significant compared to the baseline. The one exception is that CAT with dual encoders achieves the highest scores for the English → Lao pair. These results suggest that integrating any form of transliteration not only helps boost performance for lower-resource languages such as Khmer and Lao, but can also provide measurable gains for higher-resource languages like Thai.

To isolate the effects of the integration methods, we average the results over the three transliteration methods (romanization, IPA, and CAT) and report the corresponding chrF++ scores compared to the baseline in Table 5. We focus on chrF++ scores because it provides a more reliable metric for these South-East Asian languages, which do not use spaces to delimit word boundaries. Across all language directions, using a shared encoder to integrate transliterations consistently improves translation performance, with gains ranging from +0.4 to +3.4 chrF++ points over the baseline. In contrast, the Concat and Dual approaches show smaller improvements or even declines when translating from English to , with changes ranging from

[7] https://github.com/huggingface/tokenizers

System	Khm → Eng	Lao → Eng	Tha → Eng	Eng → Khm	Eng → Lao	Eng → Tha
Baseline	37.8/9.3	39.3/11.4	39.4/11.3	40.4/18.1	44.3/21.0	42.6/25.0
CAT Concat	40.0*/**11.7***	42.3*/14.0*	41.6*/12.4*	40.7*/18.4	44.7*/21.5*	43.2*/25.7*
CAT Shared	40.5*/11.5*	41.5*/12.8*	40.9*/11.9*	40.7/18.2	45.0*/21.6*	42.8/25.4
CAT Dual	39.9*/10.5*	42.6*/14.2*	42.2*/12.8*	41.4*/19.0*	**45.5***/**22.4***	43.6*/26.4*
IPA Concat	39.2*/10.7*	41.6*/13.7*	41.1*/11.4	39.0*/16.7*	43.2*/19.9*	41.0*/23.4*
IPA Shared	**42.2***/11.6*	**43.4***/**14.8***	**43.2***/**13.5***	**41.5***/**19.3***	45.4*/22.2*	**43.9***/**26.8***
IPA Dual	39.9*/10.1*	41.5*/13.4*	40.7*/12.0*	38.8*/16.1*	42.9*/19.5*	40.5*/22.6*
Rom. Concat	38.9*/10.6*	41.9*/13.7*	41.1*/12.3*	40.3/17.9	44.2/21.0	42.3/25.0
Rom. Shared	40.8*/11.0*	41.7*/13.3*	40.3*/11.6	40.1*/17.9	44.3/21.2	42.3/24.9
Rom. Dual	39.2*/10.4*	41.2*/13.9*	40.0*/11.5	39.8*/17.2*	43.3*/20.1*	41.4*/23.8*

Table 4: chrF++/BLEU scores for each transliteration method and architecture across all language directions. Scores are reported as chrF++/BLEU. Bold values indicate the best score within a language direction. An asterisk (*) marks scores that are significantly different from the Baseline ($p < 0.05$).

-0.8 to +2.6. These results highlight that the shared encoder is the most robust method for integrating transliterations for this dataset.

Focusing on the transliteration methods themselves, we average the results over the integration methods (concatenation, shared encoder, dual encoder) and report the chrF++ scores in Table 6, again comparing the averaged scores to the baseline. Unlike the integration methods, there is no single transliteration approach that consistently achieves the largest gains across all directions. IPA performs best on average when translating into English, with improvements ranging from +2.3 to +2.6 chrF++, but it underperforms when translating from English, with declines between -0.8 and -0.5. However, CAT performs best on average for English → X directions, as well as providing more consistent improvements across all language directions, with score increases ranging from +0.5 to +2.8. Romanization generally improves over the baseline but tends to achieve smaller gains than IPA or CAT.

According to these experiments, there is not a clear transliteration method which performs better than all the others. We see that both IPA and CAT enhance these MNMT models more than romanization, but not by much. Despite the much larger token overlap when using CAT, it does not do much better than the IPA performance. Though CAT results in much higher token overlap across languages, its performance is not substantially better than IPA. We hypothesize that this may be due to CAT's tendency to overgeneralize: it creates shared tokens between languages that do not necessarily share semantics, which can introduce ambiguity. Conversely, IPA enforces stricter token sharing, resulting in more precise and less ambiguous representations that facilitate effective knowledge transfer.

Both IPA and CAT provide larger improvements to the MNMT models compared to romanization, though the differences are relatively modest. Overall, all three transliteration methods contribute to improved translation, particularly in low-resource settings, despite the apparent lack of correlation to the amount of vocabulary overlap as described in Section 5.1.

Future work should investigate whether the shared tokens for each transliteration method actually preserve semantic equivalence across languages, or if their overlap introduces misleading or ambiguous representations.

6 Conclusion

Low-resource languages with unique writing systems pose challenges for traditional Neural Machine Translation (NMT) knowledge transfer techniques. In this work, we proposed methods for integrating phonetic transliterations to address the lack of shared orthographies between related high- and low-resource languages in Multilingual NMT (MNMT) systems. Specifically, we compared three transliteration schemes—International Phonetic Alphabet (IPA), romanization, and our custom Cognate-Aware Transliterations (CAT)—together with three integration methods in a Transformer model: concatenating inputs, using a shared encoder, and using dual encoders. We evaluated this methodology for Khmer, Lao, and Thai in both directions with English, leveraging knowledge transfer from the higher-resource Thai to the lower-resource Lao and Khmer.

Overall, integrating any transliteration method via any integration strategy improves translation performance in the X → English direction, while

System	Khm → Eng	Lao → Eng	Tha → Eng	Eng → Khm	Eng → Lao	Eng → Tha
Baseline	37.8 (+0.0)	39.3 (+0.0)	39.4 (+0.0)	40.4 (+0.0)	44.3 (+0.0)	42.6 (+0.0)
Concat Average	39.4 (+1.6)	41.9 (+2.6)	41.3 (+1.9)	40.0 (-0.4)	44.0 (-0.3)	42.2 (-0.4)
Shared Average	**41.2 (+3.4)**	**42.2 (+2.9)**	**41.5 (+2.1)**	**40.8 (+0.4)**	**44.9 (+0.6)**	**43.0 (+0.4)**
Dual Average	39.7 (+1.9)	41.8 (+2.5)	41.0 (+1.6)	40.0 (-0.4)	43.9 (-0.4)	41.8 (-0.8)

Table 5: chrF++ scores for the three phonetic integration methods, averaged over all transliteration methods (Romanization, IPA, CAT) compared to the Baseline. Bold values indicate the best score within a language direction. Values in parentheses indicate the change relative to the Baseline.

System	Khm → Eng	Lao → Eng	Tha → Eng	Eng → Khm	Eng → Lao	Eng → Tha
Baseline	37.8 (+0.0)	39.3 (+0.0)	39.4 (+0.0)	40.4 (+0.0)	44.3 (+0.0)	42.6 (+0.0)
CAT Average	40.1 (+2.3)	42.1 (+2.8)	41.6 (+2.2)	**40.9 (+0.5)**	**45.1 (+0.8)**	**43.2 (+0.6)**
IPA Average	**40.4 (+2.6)**	**42.2 (+2.9)**	**41.7 (+2.3)**	39.8 (-0.6)	43.8 (-0.5)	41.8 (-0.8)
Rom. Average	39.6 (+1.8)	41.6 (+2.3)	40.5 (+1.1)	40.1 (-0.3)	43.9 (-0.4)	42.0 (-0.6)

Table 6: chrF++ scores for the three transliteration methods, averaged over all integration methods (concatenated input, shared encoder, dual encoder) compared to the Baseline. Bold values indicate the best score within a language direction. Values in parentheses indicate the change relative to the Baseline.

translations from English → X show less consistent gains. Among all combinations, using a shared encoder with IPA or CAT transliterations achieves the largest improvements. Notably, the Khmer → English direction—our lowest-resource scenario—achieves the highest chrF++ improvement of +4.4 points, providing strong evidence of effective knowledge transfer between these South-East Asian languages.

This approach can be extended to other language groups that share linguistic features but not orthography, such as Maltese (Latin script) and Tunisian Arabic (Arabic script), with the potential to enhance translation for lower-resource languages. Future work could also explore additional transliteration and integration methods, as well as leverage larger datasets such as OPUS for South-East Asian languages, which would likely further improve performance above the baseline. Beyond multilingual learning for knowledge transfer, additional work could explore whether integrating transliterations benefits parent-child fine-tuning (Zoph et al., 2016; Neubig and Hu, 2018) in which a parent model is first trained on a high-resource language pair and then fine-tuned on the low-resource language pair.

Limitations

This study focuses on a single group of related languages and may not generalize to other language families containing different orthographies. All models were trained under fixed architectural conditions, and results could differ when scaling models up or down. We trained using Paracrawl Bonus data only, without incorporating additional OPUS data, in order to maintain smaller models. While this allows for controlled and informative experiments, we acknowledge that including all available data would likely improve overall translation metrics.

We note that creating a Cognate-Aware Transliteration (CAT) system requires expertise in the languages involved. Unlike IPA or romanization schemes, which are more widely available and easier to apply across languages, there is currently no automated way to generate a CAT system for a given set of languages.

References

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. 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 3874–3884, Minneapolis, Minnesota. Association for Computational Linguistics.

Alham Fikri Aji, Nikolay Bogoychev, Kenneth Heafield, and Rico Sennrich. 2020. In neural machine translation, what does transfer learning transfer? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7701–7710, Online. Association for Computational Linguistics.

Chantal Amrhein and Rico Sennrich. 2020. On Romanization for model transfer between scripts in neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages

2461–2469, Online. Association for Computational Linguistics.

Cecil H Brown, Eric W Holman, and Søren Wichmann. 2013. Sound correspondences in the world's languages. *Language*, 89(1):4–29.

Wei-Rui Chen and Muhammad Abdul-Mageed. 2021. Machine translation of low-resource Indo-European languages. In *Proceedings of the Sixth Conference on Machine Translation*, pages 347–353, Online. Association for Computational Linguistics.

Xilun Chen, Ahmed Hassan Awadallah, Hany Hassan, Wei Wang, and Claire Cardie. 2019. Multi-source cross-lingual model transfer: Learning what to share. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3098–3112, Florence, Italy. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.

Nicholas James Enfield. 2019. *Mainland Southeast Asian languages: A concise typological introduction*. Cambridge University Press.

Markus Freitag and Orhan Firat. 2020. Complete multilingual neural machine translation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 550–560, Online. Association for Computational Linguistics.

Jay Gala, Pranjal A Chitale, Raghavan AK, Varun Gumma, Sumanth Doddapaneni, Aswanth Kumar, Janki Nawale, Anupama Sujatha, Ratish Puduppully, Vivek Raghavan, and 1 others. 2023. Indictrans2: Towards high-quality and accessible machine translation models for all 22 scheduled indian languages. *arXiv preprint arXiv:2305.16307*.

Mozhdeh Gheini and Jonathan May. 2019. A universal parent model for low-resource neural machine translation transfer. *arXiv preprint arXiv:1909.06516*.

Stig-Arne Grönroos, Sami Virpioja, and Mikko Kurimo. 2018. Cognate-aware morphological segmentation for multilingual neural translation. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 386–393, Belgium, Brussels. Association for Computational Linguistics.

Ulf Hermjakob, Jonathan May, and Kevin Knight. 2018. Out-of-the-box universal Romanization tool uroman. In *Proceedings of ACL 2018, System Demonstrations*, pages 13–18, Melbourne, Australia. Association for Computational Linguistics.

Jaavid J, Raj Dabre, Aswanth M, 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, Bangkok, Thailand. Association for Computational Linguistics.

Paul Jaccard. 1901. Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bull Soc Vaudoise Sci Nat*, 37:547–579.

Philipp Koehn. 2024. Neural methods for aligning large-scale parallel corpora from the web for south and east asian languages. In *Proceedings of the ninth conference on machine translation*, pages 1454–1466.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *Preprint*, arXiv:1910.13461.

Tomasz Limisiewicz, Jiří Balhar, and David Mareček. 2023. Tokenization impacts multilingual language modeling: Assessing vocabulary allocation and overlap across languages. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5661–5681, Toronto, Canada. Association for Computational Linguistics.

Ibraheem Muhammad Moosa, Mahmud Elahi Akhter, and Ashfia Binte Habib. 2023. Does transliteration help multilingual language modeling? In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 670–685, Dubrovnik, Croatia. Association for Computational Linguistics.

Benjamin Muller, Antonios Anastasopoulos, Benoît Sagot, and Djamé Seddah. 2021. When being unseen from mBERT is just the beginning: Handling new languages with multilingual language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 448–462, Online. Association for Computational Linguistics.

Graham Neubig and Junjie Hu. 2018. Rapid adaptation of neural machine translation to new languages. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 875–880, Brussels, Belgium. Association for Computational Linguistics.

Thi-Vinh Ngo, Phuong-Thai Nguyen, Thanh-Le Ha, Khac-Quy Dinh, and Le-Minh Nguyen. 2020. Improving multilingual neural machine translation for low-resource languages: French, English - Vietnamese. In *Proceedings of the 3rd Workshop on Technologies for MT of Low Resource Languages*,

pages 55–61, Suzhou, China. Association for Computational Linguistics.

Hoang Nguyen, Chenwei Zhang, Ye Liu, Natalie Parde, Eugene Rohrbaugh, and Philip S. Yu. 2024. CORI: CJKV benchmark with Romanization integration - a step towards cross-lingual transfer beyond textual scripts. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 4008–4020, Torino, Italia. ELRA and ICCL.

Hoang Nguyen, Chenwei Zhang, Tao Zhang, Eugene Rohrbaugh, and Philip Yu. 2023. Enhancing cross-lingual transfer via phonemic transcription integration. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9163–9175, Toronto, Canada. Association for Computational Linguistics.

Hoang H Nguyen, Khyati Mahajan, Vikas Yadav, Julian Salazar, Philip S. Yu, Masoud Hashemi, and Rishabh Maheshwary. 2025. Prompting with phonemes: Enhancing LLMs' multilinguality for non-Latin script languages. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 11975–11994, Albuquerque, New Mexico. Association for Computational Linguistics.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, and 20 others. 2024. Scaling neural machine translation to 200 languages. *Nature*, 630(8018):841–846.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Vaidehi Patil, Partha Talukdar, and Sunita Sarawagi. 2022. Overlap-based vocabulary generation improves cross-lingual transfer among related languages. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 219–233, Dublin, Ireland. Association for Computational Linguistics.

Wannaphong Phatthiyaphaibun, Korakot Chaovavanich, Charin Polpanumas, Arthit Suriyawongkul, Lalita Lowphansirikul, Pattarawat Chormai, Peerat Limkonchotiwat, Thanathip Suntorntip, and Can Udomcharoenchaikit. 2023. Pythainlp: Thai natural language processing in python. In *Proceedings of the 3rd Workshop for Natural Language Processing Open Source Software (NLP-OSS 2023)*, pages 25–36.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Maja Popović. 2017. chrf++: words helping character n-grams. In *Proceedings of the second conference on machine translation*, pages 612–618.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Sukannya Purkayastha, Sebastian Ruder, Jonas Pfeiffer, Iryna Gurevych, and Ivan Vulić. 2023. Romanization-based large-scale adaptation of multilingual language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7996–8005, Singapore. Association for Computational Linguistics.

Nathaniel Romney Robinson, Matthew Dean Stutzman, Stephen D. Richardson, and David R Mortensen. 2023. African substrates rather than european lexifiers to augment african-diaspora creole translation. In *4th Workshop on African Natural Language Processing*.

Elizabeth Salesky, Neha Verma, Philipp Koehn, and Matt Post. 2023. Multilingual pixel representations for translation and effective cross-lingual transfer. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13845–13861, Singapore. Association for Computational Linguistics.

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.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 833–844, Hong Kong, China. Association for Computational Linguistics.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575, Austin, Texas. Association for Computational Linguistics.

A Cleaning Steps

We apply the following cleaning steps in our data cleaning pipeline:

1. Remove pairs containing empty source or target segments.
2. Remove pairs when the source segment exactly or nearly matches the target segment.
3. Remove duplicate source-target pairs.
4. Remove pairs with segments containing mostly non-alphabetic characters.
5. Remove pairs with segments containing abnormally long sequences of characters without spaces, including segments that are only URLs.
6. Remove pairs containing segments with unbalanced brackets.
7. Remove pairs containing fewer than 3 words in the English source segment.
8. Remove pairs with segments containing a higher number of characters than 5 standard deviations above the mean for that language (sentences that are too long).
9. Remove pairs in which the ratio of the lengths of the source and target segments exceeds a certain cutoff.
10. Normalize escaped Unicode characters.
11. Validate and normalize character encodings for each language.
12. Normalize whitespace
13. Shorten sequences of excessively repeated punctuation.
14. Normalize quotation marks.
15. Normalize HTML entities.
16. Remove all markup tags.

B Khmer, Lao, and Thai Cognate-Aware Transliteration (CAT)

Creation of a Cognate Aware Transliteration (CAT) system requires familiarity with the languages it is designed to incorporate. The ideal CAT system uses examples of known cognates to detect common, predictable mappings between phonemes across multiple languages, including both vowels and consonants. We did this manually, but finding these mappings automatically is likely possible and a topic for future research.

For Thai, Lao, and Khmer, we created these mappings based on cognates, borrowed words, and place names that could be found in both languages. Specifically, we constructed these mappings through a comparative dictionary-based approach. Each language was examined letter by letter, and for each grapheme we identified potential correspondences by consulting cognates, loanwords, and place names attested across the three languages. When a candidate word exhibited both phonological similarity and a plausible semantic match across the languages, we treated it as evidence of a sound correspondence for that grapheme. This procedure relied on the combined expertise of the researchers, who brought working knowledge of the relevant languages, ensuring that proposed correspondences were grounded in linguistic judgment. We also considered similarities in orthography when creating mappings, such as when two graphemes exhibited a large degree of visual similarity, such as when two graphemes had closely aligned visual features—length, curvature, and positioning—making them appear almost identical (e.g., Khmer vowel ◌ា and Thai vowel า).

For this example, we designed the system to maximize overlap and cognates, allowing for cognates with different romanization and pronunciations to be successfully identified. However, this may have led to the creation of false cognates, negating some of the benefits of transfer learning. In addition, because Khmer is not tonal, we chose not to map the tones between Thai and Lao for commonality. Mapping these may improve transfer learning between Thai and Lao at the cost of transfer learning between these two languages and Khmer.

To reduce complexity, we modeled cognate consonant phonemes based on beginning consonants only, but mapping final consonants would lead to a more complete CAT system. We chose not to do this because of the complexity of determining whether a consonant is beginning or final in Thai and Khmer.