Happiness is Sharing a Vocabulary: A Study of Transliteration Methods

Anonymous ACL submission

Abstract

Transliteration has emerged as a powerful means to bridge the gap between various languages in multilingual NLP, showing promising results on unseen languages without respect to script. While it is widely understood that this success is due to the degree to which transliteration results in a shared representational space among languages, we investigate the degree to which shared script, an overlap in token vocabularies, and shared phonology contribute to performance of models relying on transliteration. To investigate this question, we train and evaluate models using three kinds of transliteration (romanization, phonemic transcription, and substitution ciphers) as well as orthography. We evaluate on two downstream tasks, named entity recognition (NER) and natural language inference (NLI), yielding results largely consistent with our hypothesis—that romanization is most effective because it results in sharing of all three kinds.

1 Introduction

Multilingual language modeling has drawn significant attention from researchers seeking to cover diverse languages and promote fairness in AI. Efforts for effective multilingual language modeling include improving the performance of low-resource languages (Bharadwaj et al., 2016), dealing with tokenization fairness across languages (Ahia et al., 2023; Petrov et al., 2023; Limisiewicz et al., 2024), investigating the curse of multilinguality (Conneau et al., 2020; Wang et al., 2020; Chang et al., 2024; Blevins et al., 2024), and breaking the script barriers (Chaudhary et al., 2018; Moosa et al., 2023; J et al., 2024; Sohn et al., 2024; Ahia et al., 2024; Liu et al., 2024). One of the recent approaches that touches on all of these problems is *transliteration*—converting original forms of written text into a unified input representations with methods such as romanization or grapheme-to-phoneme (G2P) transduction.

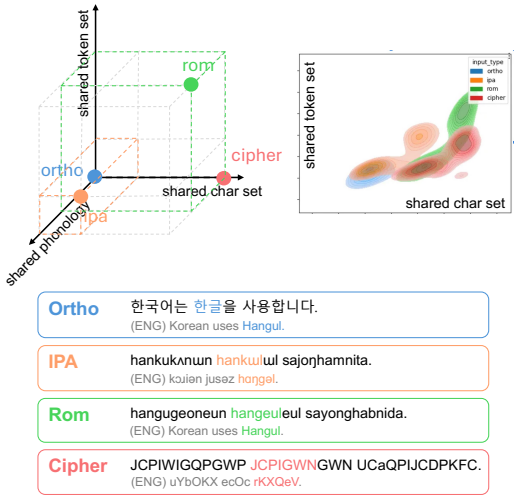

Figure 1: *Top left:* Conceptual visualization of the transliteration analysis schema, positioning input types (Ortho, IPA, Rom, Cipher) based on shared character set, token set, and phonology. *Top right:* KDE plot showing empirical distribution of overlap ratios for each quantifiable component. *Bottom:* Transliteration examples generated with each method.

Transliteration in multilingual NLP is typically performed using Latin scripts or International Phonetic Alphabet (IPA), giving various languages a shared input representation. Both representations encode linguistic information—specifically phonetic and phonological—across languages. Here, we pose a question: *Is it the shared script itself or the linguistic information encoded in the scripts that helps the models adapt to other languages?*

To investigate this question, we define three key factors in transliteration—(i) shared character set, (ii) shared token set, and (iii) shared phonology—that influence how a model processes and generalizes across languages. We then run experiments with four different input types, each varying in the degree to which these factors are present: Orthography, IPA, Romanized, and Substitution Ciphered text (see Figure 1). IPA and Romanized text encode linguistic information (phonetic or phonological)

to different extents, making them more likely to leverage shared phonology (e.g., similarity in cognate and borrowed vocabulary items) and contain shared tokens. On the other hand, ciphered text shares the same character set as romanized text but lacks any linguistic information, as each language is randomly mapped to different letters.

We hypothesize that **romanized text yields the best performance** in handling diverse languages as it improves representations across all three dimensions. Based on this assumption, IPA is expected to follow, as it enhances two out of three dimensions (shared phonology and tokens) while ciphered text only shares the character set and lacks other shared representations. Throughout the paper, we evaluate our hypothesis by comparing downstream task performance on seen and unseen languages and analyze each method in terms of token overlaps.

2 Preliminary: Transliteration for Multilingual Language Modeling

Transliteration has been recently explored as a method to enhance cross-lingual transfer in multilingual NLP by unifying script representations. Two major approaches in this domain are phonemic transcription and romanization.

Phonemic transcriptions use IPA to represent various languages. It has been explored in cross-lingual scenarios, particularly to low-resource languages (Bharadwaj et al., 2016; Chaudhary et al., 2018; Nguyen et al., 2023; Sohn et al., 2024). Recently, Nguyen et al. (2024) show that IPA prompting aids large-scale LLMs in handling non-Latin scripts. Similarly, romanization has been widely used to overcome the difference in scripts and mitigate potential out-of-vocabulary problems by restricting the input space (Fujinuma et al., 2022; Moosa et al., 2023; Liu et al., 2024). This approach improves POS Tagging and Dependency Parsing by enhancing token consistency (Fujinuma et al., 2022) and significantly benefits low-resource languages without negatively impacting high-resource ones (Moosa et al., 2023).

3 Input Types

While transliteration into shared scripts has demonstrated promising results in cross-lingual transfer, particularly for low-resource languages and non-Latin scripts (Soni and Bhattacharyya, 2024; J et al., 2024), its underlying mechanisms remain unexplored. As illustrated in Figure 1, we define three key factors that explain different aspects of transliteration.

- **Shared Character Set.** Transliteration usually enforces a shared character set across languages. For example, romanization can only produce Latin characters, which significantly reduces the number of unique characters and patterns that a tokenizer must learn.
- **Shared Token Set.** Here, we specifically distinguish *tokens* from *characters*, where by tokens we refer to subword tokens that contain more than a character.
- **Shared Phonology.** Widely used transliteration methods (e.g., G2P and romanization) encode phonological information in their representations. Representing languages based on their phonology can capture representations of cognate and borrowed vocabulary shared across languages.

To explore these different dimensions of transliteration, we employ four distinct input types: Orthography (Ortho), IPA, Romanized text (Rom), and Substitution Ciphered text (Cipher). Here, we explain in detail the process of converting written text data (Ortho) into each of other input types.

3.1 G2P Conversion (IPA)

Based on Latin scripts, IPA symbols are designed to represent pronunciations of human language in phonemes. While transliteration into IPA enables some degree of character set sharing, differences in phonemic inventories and phonotactic structures cause each language to use its own distinct set of characters and subword tokens. To convert orthographic data into IPA symbols, we use Epitran (Mortensen et al., 2018), a widely used rule-based G2P tool that supports more than a hundred languages.

3.2 Romanization (Rom)

Romanization converts various scripts into Latin alphabets, enforcing a stricter limit that enables multiple languages to share the character set. Additionally, unlike G2P, which converts identical Latin-script text into language-specific phonemes, romanization preserves the original form of text written in Latin scripts. Since Latin scripts encode sound—though not as precisely as IPA—romanization produces phonologically informed representations for each language. We employ Uroman (Hermjakob et al., 2018) which supports more than 370 languages for romanization.

3.3 Substitution Cipher (Cipher)

A substitution cipher is a method from cryptography where units of plaintext are replaced with ciphertext according to a predefined rule or key. We apply substitution cipher to the romanized text of each language—in different rules—to remove encoded phonological information. While this allows multilingual text to share the same character space as Rom, it no longer contains phonological meanings and prevents the sharing of meaningful subword tokens across languages. We employ Caesar cipher, a simple substitution encryption technique. Details are provided in Appendix A.4.

4 Experiments

4.1 Language Selection

	Script	
	same	diverse
similar	swe, por, lij, cat, ron, spa, sqi, fra	fra, ben, hin, hrv, ori, rus, srp, urd
dissimilar	ilo, sna, lav, uzb, deu, fin, som, swa	amh, ben, tel, fra, tha, kat, kor, mya

Table 1: Languages selected for each language set.

To examine how different input types impact multilingual adaptation, we selected languages to form four language sets: (i) typologically similar languages using the same script (sim-same), (ii) similar languages using diverse scripts (sim-div), (iii) dissimilar languages using the same script (dissim-same), and (iv) dissimilar languages using diverse scripts (dissim-div). Similar to Chang et al. (2024), we utilized lang2vec (Littell et al., 2017)[1] to compute language similarity. We extracted syntactic, geographic, and genetic features from lang2vec to obtain cosine similarities, and also defined lexical similarity based on word overlap ratio between training corpora of each language[2]. By aggregating these similarity scores, as detailed in Appendix A.1, we assigned eight languages to each set (see Table 1) and trained multilingual models with varying linguistic similarities and scripts.

4.2 Datasets

For pre-training, we utilize sampled version of a preprocessed Wikipedia corpus from Hugging Face.[3] For downstream task, we utilized WikiAnn (Pan et al., 2017; Rahimi et al., 2019) dataset for NER and XNLI (Conneau et al., 2018) for sentence classification (NLI) task. More details on preprocessing and dataset statistics can be found in Appendix A.9. In order to train the model with different input types, we converted all datasets into each input type.

4.3 Model Training

To investigate the impact of different input types, we pre-train 16 models from scratch using four input types and four language sets. We avoid using publicly available pre-trained models to ensure a controlled experimental setup, as most such models are optimized for orthography, preventing fair comparison across transliteration methods.

We first trained a SentencePiece (character-level) BPE subword tokenizer for each model with fixed vocabulary size of 30K for all tokenizers. We employed a Transformer architecture, following the training regime of RoBERTa (Liu et al., 2019) with masked language modeling on a multilingual corpus. After pre-training we fine-tuned each model on target language dataset to obtain downstream task performance. For details on the model configurations and training, refer to Appendix A.2 and Appendix A.3.

5 Results: Downstream Task Performance across Input Types

Table 2 presents average scores across target languages for downstream tasks. Average F1 scores of each model for seen and unseen languages are provided for NER,[4] and average accuracies for XNLI. p-values obtained from paired t-tests on F1 scores across different input types can be found in Appendix A.5.

NER Performance in Seen/Unseen Languages. Transliteration does not provide a significant advantage over orthographic text when the language was seen during pre-training. While Rom outperforms other input types including Ortho, its superiority over Ortho is not statistically significant ($p > 0.05$). On the other hand, for unseen languages, the performance of Ortho is significantly lower than that of all other input types ($p < 0.05$). Furthermore, we find that our hypothesis holds, with Rom achieving the highest average F1 scores in 6 out of 8 cases. Interestingly, IPA and Cipher do not show statistically significant differences, despite Cipher containing no shared linguistic information. We further investigate this in Section 6.

[1] Utilizing https://github.com/antonisa/lang2vec

[2] Words are segmented by white spaces.

[3] https://huggingface.co/datasets/wikimedia/wikipedia

[4] Unseen languages refer to languages not included in pre-training of each model.

Trained Lang. Set	Named Entity Recognition								XNLI			
	Seen				Unseen							
	Ortho	IPA	Rom	Cipher	Ortho	IPA	Rom	Cipher	Ortho	IPA	Rom	Cipher
sim-same	**0.8466**	0.8085	0.8395	0.8173	0.6611	0.6801	**0.7267**	0.6824	0.5793	0.6045	**0.6276**	0.6137
sim-div	0.8409	0.8239	**0.8451**	0.8270	0.6321	0.6787	**0.7151**	0.6772	0.6007	0.6135	**0.6224**	0.6096
dissim-same	0.7860	0.7732	**0.7981**	0.7725	0.6626	0.7468	0.7280	**0.7547**	0.5971	0.6087	**0.6137**	0.5972
dissim-div	0.7402	0.7524	**0.7538**	0.7518	0.7450	0.7524	**0.7832**	0.7496	0.5860	0.6214	**0.6327**	0.6254

Table 2: Downstream task performances averaged across target languages—F1 scores for NER and accuracy scores for XNLI. **Bold**: best performing input. Underlined: second best.

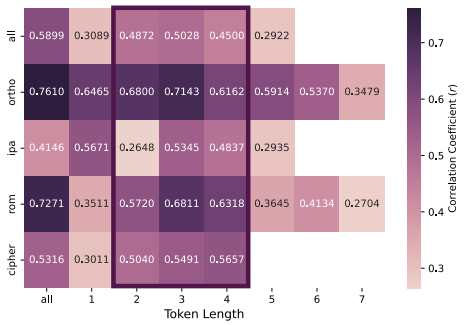

Figure 2: Pearson r between overlap ratios of each token length and NER performance. Correlations with $p > 0.05$ are masked out.

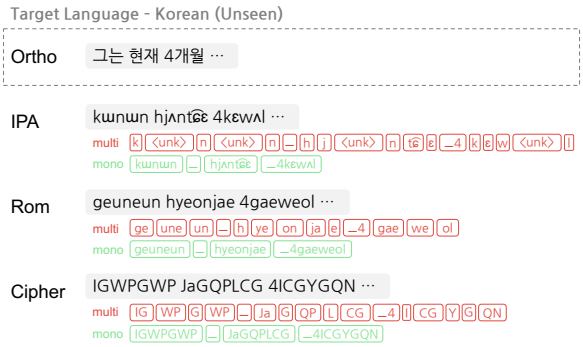

Figure 3: Tokenization results for an incomplete Korean sentence (English: "He is currently 4 months..."). Red indicates multilingual tokenizers (trained on sim-div), whereas light green shows monolingual Korean tokenizers, serving as an ideal reference.

XNLI Performance. For XNLI, we did not distinguish between seen and unseen languages due to the limited number of supported languages. Instead, we randomly sampled 10 languages from the supported languages and fine-tuned on each to obtain accuracy scores. The trend was consistent across all models: Rom outperformed the others, while IPA and Cipher demonstrated comparable performance.

6 Analysis: Shared Tokens

Transliteration is widely assumed to enhance multilingual language modeling by increasing token overlap. However, it is not clear whether it is driven by the same script itself or the consistent linguistic information encoded in the script. To investigate this question, we analyze length-wise lexical overlap each transliteration method produces, as defined in Appendix A.6.

Token Overlap and Transferability. Figure 2 presents the Pearson correlation coefficient between overlap ratios and NER performance for each input type. We observe that sharing tokens with trained languages is crucial for successful adaptation to unseen languages. Particularly, token lengths of 2 to 4 exhibit a strong correlation with F1 scores, highlighting the importance of sharing short character sequences as subword patterns across languages. We additionally provide a box plot in Figure 5, which shows overlap ratios of each input type by token length.

Comparison between IPA and Cipher. While both IPA and Cipher perform better than Ortho on unseen languages, they are suboptimal compared to Rom. IPA represents phonological information, allowing for shared character sequences across languages, such as those capturing common syllable structures. However, phonemic transcription reflects language-specific phonological inventories, hindering a shared character set and thereby causing unknown tokens ([UNK]) (See Figure 3). On the other hand, Cipher shares a character set, but each character encodes no linguistic information common across languages. Yet, sharing characters allows the model to adapt token embeddings, resulting in performance comparable to IPA.

By comparing IPA and Cipher, we disentangle the roles of linguistic information and character sharing, observing that both contribute to transfer to unseen languages. This supports the effectiveness of Rom, which combines both properties and yields more transferable shared tokens.

7 Limitation

The results reported here are suggestive, but there are three major limitations which prevent us from generalizing them too broadly. First, we only tested one type of transformer model with one tokenization scheme. It is possible, for example, that we would have obtained much different results if we had trained character- or byte-level models. Also, we only tested one romanizer and one G2P transducer. It is entirely possible that we would have obtained different results if different tools had been used.

8 Ethics Statement

We believe that this research raises no significant ethical concerns or violations of the code of ethics mandated by the Association for Computational Linguistics. The data used in this study, all of which are publicly available, were collected in accordance with legal and institutional protocols, to the best of our knowledge. Furthermore, our use of these resources is compatible with the uses intended by the creators.

References

Orevaoghene Ahia, Sachin Kumar, Hila Gonen, Valentin Hofmann, Tomasz Limisiewicz, Yulia Tsvetkov, and Noah A. Smith. 2024. MAGNET: Improving the multilingual fairness of language models with adaptive gradient-based tokenization. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Orevaoghene Ahia, Sachin Kumar, Hila Gonen, Jungo Kasai, David Mortensen, Noah Smith, and Yulia Tsvetkov. 2023. Do all languages cost the same? tokenization in the era of commercial language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9904–9923, Singapore. Association for Computational Linguistics.

Akash Bharadwaj, David Mortensen, Chris Dyer, and Jaime Carbonell. 2016. Phonologically aware neural model for named entity recognition in low resource transfer settings. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1462–1472, Austin, Texas. Association for Computational Linguistics.

Terra Blevins, Tomasz Limisiewicz, Suchin Gururangan, Margaret Li, Hila Gonen, Noah A. Smith, and Luke Zettlemoyer. 2024. Breaking the curse of multilinguality with cross-lingual expert language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10822–10837, Miami, Florida, USA. Association for Computational Linguistics.

Tyler A. Chang, Catherine Arnett, Zhuowen Tu, and Ben Bergen. 2024. When is multilinguality a curse? language modeling for 250 high- and low-resource languages. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4074–4096, Miami, Florida, USA. Association for Computational Linguistics.

Aditi Chaudhary, Chunting Zhou, Lori Levin, Graham Neubig, David R. Mortensen, and Jaime Carbonell. 2018. Adapting word embeddings to new languages with morphological and phonological subword representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3285–3295, Brussels, Belgium. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard 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, Online. Association for Computational Linguistics.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Yoshinari Fujinuma, Jordan Boyd-Graber, and Katharina Kann. 2022. Match the script, adapt if multilingual: Analyzing the effect of multilingual pretraining on cross-lingual transferability. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1500–1512, Dublin, Ireland. 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.

David Kahn. 1996. *The Codebreakers: The Comprehensive History of Secret Communication from Ancient*

Times to the Internet, revised edition. Scribner, New York.

Tomasz Limisiewicz, Terra Blevins, Hila Gonen, Orevaoghene Ahia, and Luke Zettlemoyer. 2024. MYTE: Morphology-driven byte encoding for better and fairer multilingual language modeling. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15059–15076, Bangkok, Thailand. Association for Computational Linguistics.

Patrick Littell, David R. Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner, and Lori Levin. 2017. URIEL and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 8–14, Valencia, Spain. Association for Computational Linguistics.

Yihong Liu, Chunlan Ma, Haotian Ye, and Hinrich Schuetze. 2024. TransliCo: A contrastive learning framework to address the script barrier in multilingual pretrained language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2476–2499, Bangkok, Thailand. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

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.

David R. Mortensen, Siddharth Dalmia, and Patrick Littell. 2018. Epitran: Precision G2P for many languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Hoang Nguyen, Khyati Mahajan, Vikas Yadav, Philip S. Yu, Masoud Hashemi, and Rishabh Maheshwary. 2024. Prompting with phonemes: Enhancing llm multilinguality for non-latin script languages. *Preprint*, arXiv:2411.02398.

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.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1958, Vancouver, Canada. Association for Computational Linguistics.

Aleksandar Petrov, Emanuele La Malfa, Philip Torr, and Adel Bibi. 2023. Language model tokenizers introduce unfairness between languages. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Afshin Rahimi, Yuan Li, and Trevor Cohn. 2019. Massively multilingual transfer for NER. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 151–164, Florence, Italy. Association for Computational Linguistics.

Jimin Sohn, Haeji Jung, Alex Cheng, Jooeon Kang, Yilin Du, and David R Mortensen. 2024. Zero-shot cross-lingual NER using phonemic representations for low-resource languages. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13595–13602, Miami, Florida, USA. Association for Computational Linguistics.

Govind Soni and Pushpak Bhattacharyya. 2024. RoMantra: Optimizing neural machine translation for low-resource languages through Romanization. In *Proceedings of the 21st International Conference on Natural Language Processing (ICON)*, pages 157–168, AU-KBC Research Centre, Chennai, India. NLP Association of India (NLPAI).

Zirui Wang, Zachary C. Lipton, and Yulia Tsvetkov. 2020. On negative interference in multilingual models: Findings and a meta-learning treatment. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4438–4450, Online. Association for Computational Linguistics.

A Appendix

A.1 Language Selection

To examine the impact on multilingual adaptation that differences in input types have, we selected four language sets : (i) similar languages using the same script (sim-same), (ii) similar languages using diverse scripts (sim-div), (iii) dissimilar languages using the same script (dissim-same), and (iv) dissimilar languages using diverse scripts (dissim-div). These sets were used to train multilingual models with varying linguistic similarities and scripts. For each set, we assigned eight languages based on a computed similarity score as shown in Table 1.

Similar to Chang et al. (2024), we utilized lang2vec (Littell et al., 2017)[5] to compute language

[5]Utilizing https://github.com/antonisa/lang2vec

similarity. Specifically, we extracted syntactic, geographic, and genetic features from lang2vec and computed cosine similarities, denoted as s_{syn}, s_{geo}, and s_{gen} in Eq. 1. We also defined lexical similarity s_{lex}, which is obtained by calculating the word overlap ratio between training corpora of each language[6]. Finally, we aggregated all similarity scores (i.e., syntactic, geographic, genetic, and lexical) to derive the overall similarity score between two languages:

$$\begin{aligned} \text{sim}_s(x, y) = \text{s}_{syn}(x, y) + \text{s}_{geo}(x, y) \\ + \text{s}_{gen}(x, y) + \text{s}_{lex}(x, y). \end{aligned} \tag{1}$$

With initial set of languages L that are supported by Wikipedia corpus and Epitran, we use average pairwise similarity scores to compute similarity score for a set of languages and obtain an optimal set L_s^*, where $s \in \{\text{sim-same}, \text{sim-div}\}$:

$$\begin{aligned} L_s^* = \arg \max_{\substack{L_s \subset L \\ |L_s|=8}} \Bigg(\frac{1}{|L_s|(|L_s|-1)} \sum_{x \in L_s} \sum_{\substack{y \in L_s \\ y \neq x}} \text{sim}_s(x, y) \\ + \alpha \cdot \Big(\mathbb{1}_{s \in \{\text{sim-div}\}} |\text{SC}_{L_s}| \\ - \mathbb{1}_{s \in \{\text{dissim-div}\}} |\text{SC}_{L_s}| \Big) \Bigg), \end{aligned} \tag{2}$$

.

As for an optimal set L_d^*, where $d \in \{\text{dissim-same}, \text{dissim-div}\}$:

$$\begin{aligned} L_d^* = \arg \min_{\substack{L_d \subset L \\ |L_d|=8}} \Bigg(\frac{1}{|L_d|(|L_d|-1)} \sum_{x \in L_d} \sum_{\substack{y \in L_d \\ y \neq x}} \text{sim}_s(x, y) \\ + \alpha \cdot \Big(\mathbb{1}_{d \in \{\text{sim-div}\}} |\text{SC}_{L_d}| \\ - \mathbb{1}_{d \in \{\text{dissim-div}\}} |\text{SC}_{L_d}| \Big) \Bigg). \end{aligned} \tag{3}$$

To select languages for the sets with same script (i.e., sim-same and dissim-same), we limited the search space to languages that use the Latin script to maximize the number of languages available for similarity-based sampling.

For sets with diverse scripts (i.e., -div), we additionally consider how many different scripts are involved in each set.

A.2 Model Configuration

Table 3 summarizes the key configuration details of our RoBERTa-based model. Number of parameters per model is 109,082,112.

[6]Words are segmented by white spaces.

Parameter	Value
Vocabulary Size	30,000
Hidden Size	768
Hidden Layers	12
Attention Heads	12
Intermediate Size	3072
Activation Function	GELU
Dropout (Hidden/Attention)	0.1
Max Position Embeddings	514

Table 3: Model Configuration

A.3 Training Setup

To investigate the impact of different input types, we pre-trained and fine-tuned a total of 16 models across four distinct input types and language sets. In addition, we trained a SentencePiece BPE tokenizer for each model, fixing the vocabulary size to 30K. Table 4 summarizes the key hyperparameters used in our experiments for both the pretraining phase and the downstream NER task.

Hyperparameter Sweep We conducted grid search to find learning rates that converges or achieves the best results. For pre-training, the search space was {1e-5, 2e-5, 3e-5, 5e-5, 1e-4, 2e-4, 3e-4} and for NER, it was {3e-5, 5e-5, 1e-4}.

Parameter	Pretraining	NER Task
FP16 Training	True	True
Max Sequence Length	512	512
Batch Size (per device)	64	64
Gradient Accumulation Steps	1	-
Warmup Steps	50	-
Learning Rate	1e-4	5e-5
Weight Decay	0.01	0.01
LR Scheduler Type	Linear	-
MLM Probability	0.15	-
Epochs	300	20
Log Interval	-	1
GPU Resources	4 NVIDIA L40S	2 NVIDIA RTX A6000

Table 4: Training Configurations

A.4 Substitution Cipher (Cipher)

A substitution cipher is a method from cryptography where units of plaintext are replaced with ciphertext according to a predefined rule or key. We apply substitution cipher to the Romanized text to remove encoded phonological information.

Specifically, we use the Caesar cipher (Kahn, 1996), a simple substitution encryption technique that shifts each letter in the text by a fixed number of positions in the Latin alphabet. For each language, we assign an integer that determines the shift from the current position of each letter. For

example, if English is assigned the integer 4, the word 'apple' would be represented as 'ettpi', with each letter replaced by the one four positions ahead in the alphabet.

A.5 P-values of Paired t-tests

Table 2 presents the NER scores for different input types across various language settings. To assess the significance of the observed differences, we performed paired t-tests. Figure 4 displays the corresponding P-values derived from these tests.

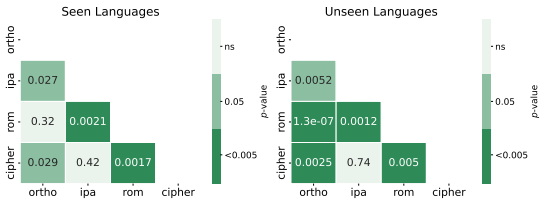

Figure 4: P-value for paired t-test on NER scores across different input types.

A.6 Lexical Overlap

We measure lexical overlap of an unseen target language by taking the overlap ratios, as defined in l_t as follows:

$$\text{Lexical Overlap}(l_t) = \max_{l_s \in L_s} \frac{\sum_{x \in (B_{l_s} \cap B_{l_t})_{\text{unique}}} f_{B_{l_t}}(x)}{|B_{l_t}|} \quad (4)$$

where l_t is a target language, l_s is one of the pre-trained languages L_s, B_l is a multiset (or bag) of subword tokens of a dataset of language l, and $f_B(x)$ is number of element x in multiset B.

A.7 Lexical Overlap of Each Input Type

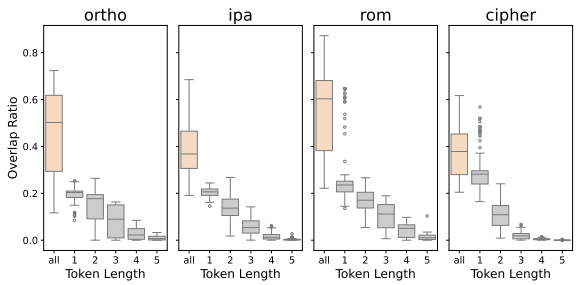

Figure 5: Distribution of lexical overlap across token lengths for different input types.

A.8 External Tools for Transliteration

In this study, we used Epitran and Uroman as transliteration tools to unify script and facilitate multilingual processing. These tools are widely used for converting text into standardized phonemic or Romanized forms, which aids in cross-lingual learning and transferability. Below, we describe their functionalities and implementation details.

Epitran(Mortensen et al., 2018) is a tool for grapheme-to-phoneme (G2P) conversion, capable of converting text into the International Phonetic Alphabet (IPA) representations. It can be downloaded from the link below https://github.com/dmort27/epitran

Uroman(Hermjakob et al., 2018) is a universal transliteration tool that converts text from various scripts into a Romanized format. It can be downloaded from the link below https://github.com/isi-nlp/uroman

A.9 Datasets

In Table 5, the specific number of datasets per corresponding language is provided. For pre-training, we utilized sampled version of preprocessed Wikipedia corpus from Huggingface[7].

We limited each language with its number of words around 10M[8]. For those languages with less number of tokens than 10M, we kept all the documents and oversampled during training, to match the model's exposure to all languages. For downstream task, we utilized WikiAnn (Pan et al., 2017; Rahimi et al., 2019) dataset for named entity recognition and XNLI (Conneau et al., 2018) for natural language inference. In order to train the model with different input types, we converted all datasets into each corresponding input type.

Wikipedia corpora used for pre-training are licensed under the GNU Free Documentation License (GFDL) and the Creative Commons Attribution-Share-Alike 3.0 License. License type for WikiAnn dataset is ODC-BY.

A.10 Detailed Experimental Results

Tables 6, 7, 8, 9 summarize the performance results (F1 scores) across different language sets under various evaluation settings. In our experiments, "Seen" refers to languages included in both pre-training and fine-tuning, "Unseen" to those entirely

[7] https://huggingface.co/datasets/wikimedia/wikipedia

[8] For each language, we randomly shuffled the order of the documents, and iterated over each document, counting the words segmented by whitespaces. We stop adding the documents when adding the number of words of the last document exceeds 10M.

absent during training. Detailed results for each setting are provided in the respective tables.

Lang	Dataset	# Train	# Validate	# Test	Lang	Dataset	# Train	# Validate	# Test
amh	wikipedia	5328	-	-	mya	wikipedia	34309	-	-
	wikiann	100	100	100		wikiann	100	100	100
ara	wikipedia	-	-	-	ori	wikipedia	11018	-	-
	wikiann	20000	10000	10000		wikiann	100	100	100
ben	wikipedia	28496	-	-	pol	wikipedia	-	-	-
	wikiann	10000	1000	1000		wikiann	20000	10000	10000
cat	wikipedia	26031	-	-	por	wikipedia	26510	-	-
	wikiann	20000	10000	10000		wikiann	20000	10000	10000
ceb	wikipedia	22724	-	-	ron	wikipedia	28890	-	-
	wikiann	100	100	100		wikiann	20000	10000	10000
deu	wikipedia	30460	-	-	rus	wikipedia	32636	-	-
	wikiann	20000	10000	10000		wikiann	20000	10000	10000
spa	wikipedia	25727	-	-	sin	wikipedia	23084	-	-
	wikiann	20000	10000	10000		wikiann	100	100	100
fin	wikipedia	36190	-	-	som	wikipedia	5204	-	-
	wikiann	20000	10000	10000		wikiann	100	100	100
fra	wikipedia	25353	-	-	sqi	wikipedia	27406	-	-
	wikiann	20000	10000	10000		wikiann	5000	1000	1000
hin	wikipedia	25492	-	-	srp	wikipedia	29961	-	-
	wikiann	5000	1000	1000		wikiann	20000	10000	10000
hrv	wikipedia	30764	-	-	swe	wikipedia	29839	-	-
	wikiann	20000	10000	10000		wikiann	20000	10000	10000
ilo	wikipedia	5828	-	-	swa	wikipedia	25911	-	-
	wikiann	100	100	100		wikiann	1000	1000	1000
kat	wikipedia	33713	-	-	tel	wikipedia	28543	-	-
	wikiann	10000	10000	10000		wikiann	1000	1000	1000
kor	wikipedia	38885	-	-	tha	wikipedia	76083	-	-
	wikiann	20000	10000	10000		wikiann	20000	10000	10000
lij	wikipedia	4002	-	-	urd	wikipedia	23568	-	-
	wikiann	100	100	100		wikiann	20000	1000	1000
lat	wikipedia	32836	-	-	uzb	wikipedia	29833	-	-
	wikiann	10000	10000	10000		wikiann	1000	1000	1000
lav	wikipedia	31152	-	-	-	-	-	-	-
	wikiann	10000	10000	10000		-	-	-	-

Table 5: Statistic of transliterated dataset. All dataset exist in four parallel versions ; original Orthographic, phonemic IPA, Romanized, and Cipher transcribed version. - refers to unavailable values. The wikipedia dataset is used for pre-training without validation or test. Languages ‘ar’ and ‘pl’ do not have available wikipedia dataset for pre-train.

		Monolingual				Multilingual			
		Ortho	IPA	Rom	Cip	Ortho	IPA	Rom	Cip
Seen	cat	0.9117	0.8970	0.9005	0.9024	0.8997	0.8725	0.8993	0.8803
	spa	0.8929	0.8759	0.8802	0.9141	0.8773	0.8584	0.8788	0.8657
	fra	0.8779	0.8607	0.8717	0.8693	0.8628	0.8252	0.8639	0.8384
	lij	0.3269	0.2927	0.4306	0.2775	0.5064	0.4052	0.4615	0.4082
	por	0.8931	0.8842	0.8891	0.8850	0.8798	0.8605	0.8796	0.8674
	ron	0.9143	0.9106	0.9153	0.9141	0.9129	0.8855	0.9103	0.8956
	sqi	0.9052	0.8958	0.9011	0.8981	0.9120	0.8738	0.8979	0.8785
	swe	0.9300	0.9238	0.9320	0.9311	0.9215	0.8872	0.9247	0.9046
Unseen	amh	-	-	-	-	0.2000	0.3089	0.3383	0.3623
	ben	-	-	-	-	0.8230	0.8907	0.9081	0.8969
	deu	-	-	-	-	0.8204	0.7400	0.8236	0.7676
	fin	-	-	-	-	0.8573	0.8050	0.8609	0.8237
	hin	-	-	-	-	0.7395	0.8043	0.8225	0.7861
	hrv	-	-	-	-	0.8682	0.8318	0.8727	0.8403
	ilo	-	-	-	-	0.6400	0.5714	0.6757	0.4498
	kat	-	-	-	-	0.6878	0.7920	0.8227	0.7780
	kor	-	-	-	-	0.5329	0.7578	0.7883	0.7626
	lav	-	-	-	-	0.8940	0.8463	0.8919	0.8695
	mya	-	-	-	-	0.2286	0.2541	0.2857	0.2232
	ori	-	-	-	-	0.2738	0.2647	0.3492	0.3533
	rus	-	-	-	-	0.8083	0.7842	0.8268	0.8010
	sna	-	-	-	-	-	-	-	-
	so	-	-	-	-	0.6256	0.4641	0.5500	0.4397
	srp	-	-	-	-	0.8574	0.8442	0.8879	0.8691
	swa	-	-	-	-	0.8250	0.7381	0.8195	0.7494
	tel	-	-	-	-	0.3384	0.5336	0.5797	0.5252
	tha	-	-	-	-	0.4762	0.6637	0.6622	0.6477
	urd	-	-	-	-	0.9032	0.9101	0.9273	0.9172
	uzb	-	-	-	-	0.8266	0.7962	0.8402	0.7862

Table 6: Performance results (F1 scores) on the sim-same language set, which consists of typologically similar languages that share the same script. The table reports results for three evaluation settings. **Seen**: languages used during both pretraining and fine-tuning, **Unseen**: languages not encountered during training and **Zero-Shot**: languages evaluated without any task-specific fine-tuning. Results are provided for four different input types: Orthographic (Ortho), IPA, Romanized (Rom), and Ciphered (Cip)

		Monolingual				Multilingual			
		Ortho	IPA	Rom	Cip	Ortho	IPA	Rom	Cip
Seen	ben	0.9584	0.9543	0.9524	0.9506	0.9380	0.9375	0.9466	0.9377
	fra	0.8779	0.8607	0.8717	0.8693	0.8436	0.8255	0.8430	0.8378
	hin	0.8909	0.8695	0.8890	0.8877	0.8524	0.8577	0.8394	0.8314
	hrv	0.8986	0.8876	0.8931	0.8950	0.8741	0.8527	0.8767	0.8605
	ori	0.6032	0.6584	0.6235	0.6721	0.5483	0.4981	0.5873	0.4962
	rus	0.8614	0.8515	0.8604	0.8578	0.8395	0.8286	0.8375	0.8304
	srp	0.9099	0.8413	0.9175	0.9117	0.8918	0.8484	0.8969	0.8900
	urd	0.9447	0.9410	0.9476	0.9408	0.9396	0.9424	0.9333	0.9318
Unseen	amh	-	-	-	-	0.0079	0.2902	0.3282	0.2695
	deu	-	-	-	-	0.7934	0.7381	0.8047	0.7608
	spa	-	-	-	-	0.8511	0.8144	0.8573	0.8265
	fin	-	-	-	-	0.8427	0.7993	0.8460	0.8201
	ilo	-	-	-	-	0.5333	0.5356	0.5537	0.4627
	ka	-	-	-	-	0.5860	0.7961	0.8162	0.7872
	kor	-	-	-	-	0.5244	0.7318	0.7792	0.7577
	lij	-	-	-	-	0.3071	0.3684	0.2975	0.3064
	lav	-	-	-	-	0.8826	0.8468	0.8891	0.8605
	mya	-	-	-	-	0.1596	0.1721	0.2975	0.2424
	por	-	-	-	-	0.8535	0.8206	0.8547	0.8312
	ron	-	-	-	-	0.8889	0.8754	0.8963	0.8695
	sna	-	-	-	-	-	-	-	-
	som	-	-	-	-	0.4874	0.4870	0.5128	0.5236
	sqi		-	-	-	0.8557	0.8319	0.8604	0.8315
	swe	-	-	-	-	0.9059	0.8583	0.9043	0.8850
	swa	-	-	-	-	0.7634	0.7429	0.7955	0.7359
	tel	-	-	-	-	0.3297	0.5753	0.6440	0.5119
	tha	-	-	-	-	0.3531	0.6680	0.6479	0.6302
	uzb	-	-	-	-	0.8384	0.7819	0.8360	0.7863

Table 7: Performance results (F1 scores) on the sim-div language set, which comprises similar languages that use diverse scripts. The table reports results for three evaluation settings. **Seen**: languages used during both pretraining and fine-tuning, **Unseen**: languages not encountered during training and **Zero-Shot**: languages evaluated without any task-specific fine-tuning. Results are provided for four different input types: Orthographic (Ortho), IPA, Romanized (Rom), and Ciphered (Cip)

		Monolingual				Multilingual			
		Ortho	IPA	Rom	Cip	Ortho	IPA	Rom	Cip
Seen	deu	0.8716	0.8518	0.8599	0.8622	0.8184	0.7924	0.8248	0.8095
	fin	0.8855	0.8813	0.8850	0.8861	0.8618	0.8264	0.8638	0.8436
	ilo	0.6053	0.6216	0.6881	0.6996	0.6757	0.7123	0.6368	0.6549
	lav	0.9284	0.9205	0.9232	0.9230	0.8995	0.8736	0.9006	0.8998
	sna	-	-	-	-	-	-	-	-
	som	0.6111	0.5648	0.6000	0.5249	0.5551	0.5887	0.6577	0.5556
	swa	0.8481	0.8385	0.8532	0.8481	0.8291	0.7981	0.8421	0.8125
	uzb	0.8648	0.8655	0.8665	0.8836	0.8621	0.8210	0.8608	0.8314
Unseen	amh	-	-	-	-	0.2833	0.5560	0.2845	0.5018
	ben	-	-	-	-	0.8269	0.8791	0.9005	0.9430
	cat	-	-	-	-	0.8733	0.8255	0.8750	0.8542
	spa	-	-	-	-	0.8518	0.8103	0.8583	0.8377
	fra	-	-	-	-	0.8312	0.7607	0.8294	0.8447
	hin	-	-	-	-	0.7128	0.8210	0.8055	0.7981
	hrv	-	-	-	-	0.8531	0.8404	0.8532	0.8495
	kat	-	-	-	-	0.6289	0.8577	0.8103	0.8606
	kor	-	-	-	-	0.5282	0.8297	0.7652	0.8381
	lij	-	-	-	-	0.3319	0.2893	0.3333	0.2979
	mya	-	-	-	-	0.2128	0.5263	0.2785	0.5750
	ori	-	-	-	-	0.0708	0.4082	0.3851	0.2339
	por	-	-	-	-	0.8566	0.8015	0.8558	0.8449
	ron	-	-	-	-	0.8906	0.8548	0.8880	0.8768
	rus	-	-	-	-	0.7992	0.7922	0.8132	0.8051
	sqi	-	-	-	-	0.8658	0.8120	0.8627	0.8259
	srp	-	-	-	-	0.8540	0.8201	0.8790	0.8739
	swe	-	-	-	-	0.9075	0.8484	0.9076	0.8919
	tel	-	-	-	-	0.3278	0.7441	0.5494	0.7632
	tha	-	-	-	-	0.5162	0.6841	0.6320	0.6110
	urd	-	-	-	-	0.8906	0.9208	0.9205	0.9220

Table 8: Performance results (F1 scores) on the dissim-same language set, which comprises typologically dissimilar languages that share the same script. The table reports results for three evaluation settings. **Seen**: languages used during both pretraining and fine-tuning, **Unseen**: languages not encountered during training and **Zero-Shot**: languages evaluated without any task-specific fine-tuning. Results are provided for four different input types: Orthographic (Ortho), IPA, Romanized (Rom), and Ciphered (Cip)

		Monolingual				Multilingual			
		Ortho	IPA	Rom	Cip	Ortho	IPA	Rom	Cip
Seen	amh	0.4796	0.4615	0.5388	0.5203	0.4941	0.5403	0.5760	0.5364
	ben	0.9584	0.9100	0.9524	0.9506	0.9579	0.9488	0.9552	0.9479
	fra	0.8779	0.8607	0.8717	0.8693	0.8528	0.8265	0.8487	0.8432
	kat	0.8866	0.8873	0.8850	0.8837	0.8647	0.8607	0.8619	0.8598
	kor	0.8611	0.8576	0.8623	0.8628	0.7699	0.8347	0.8382	0.8333
	mya	0.5401	0.5852	0.5617	0.5188	0.5259	0.5738	0.5164	0.5477
	tel	0.7880	0.7983	0.7822	0.7922	0.7532	0.7529	0.7528	0.7734
	tha	0.7052	0.6880	0.6656	0.6726	0.7031	0.6813	0.6810	0.6727
Unseen	cat	-	-	-	-	0.8797	0.8503	0.8803	0.8513
	deu	-	-	-	-	0.8088	0.7555	0.8134	0.7855
	spa	-	-	-	-	0.8615	0.8315	0.8687	0.8352
	fin	-	-	-	-	0.8504	0.8188	0.8532	0.8311
	hin	-	-	-	-	0.6585	0.8223	0.8472	0.7939
	hrv	-	-	-	-	0.8642	0.8381	0.8652	0.8428
	ilo	-	-	-	-	0.5272	0.5726	0.5122	0.4516
	lij	-	-	-	-	0.3465	0.3243	0.3793	0.2833
	lav	-	-	-	-	0.8948	0.8544	0.891	0.8762
	ori	-	-	-	-	0.3840	0.3931	0.4373	0.2913
	por	-	-	-	-	0.8609	0.8245	0.8630	0.8437
	ron	-	-	-	-	0.8940	0.8746	0.8979	0.8788
	rus	-	-	-	-	0.6753	0.7941	0.8207	0.8049
	sna	-	-	-	-	-	-	-	-
	som	-	-	-	-	0.6140	0.4893	0.5462	0.5299
	sqi	-	-	-	-	0.8720	0.8395	0.8533	0.8389
	srp	-	-	-	-	0.6697	0.8405	0.8826	0.8735
	swe	-	-	-	-	0.9117	0.8641	0.9119	0.8920
	swa	-	-	-	-	0.7968	0.7527	0.7855	0.7536
	urd	-	-	-	-	0.6974	0.9072	0.9243	0.9226
	uzb	-	-	-	-	0.8317	0.8004	0.8300	0.8121

Table 9: Performance results (F1 scores) on the dissim-div language set, which comprises typologically dissimilar languages that utilize diverse scripts. The table reports results for two evaluation settings. **Seen**: languages used during both pretraining and fine-tuning and **Unseen**: languages not encountered during training. Zero-shot evaluation was omitted due to the minimal shared representations among dissim-div languages, which limits the effectiveness of zero-shot transfer. Results are provided for four different input types: Orthographic (Ortho), IPA, Romanized (Rom), and Ciphered (Cip)