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 use named entity recognition as a downstream task for evaluation. Our results are 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.

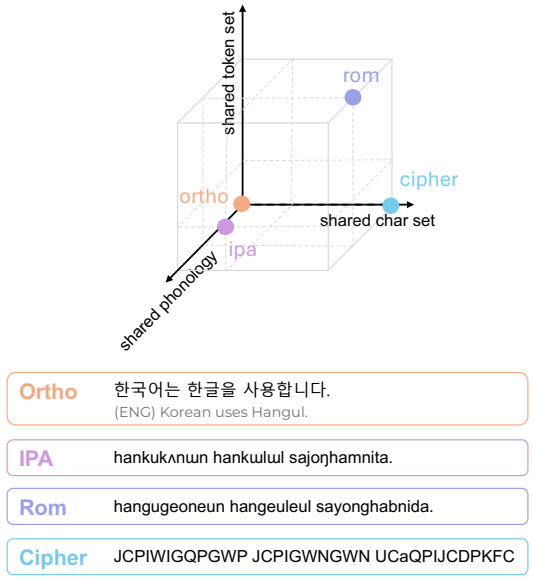


Figure 1: Visualization of transliteration analysis schema, showing input types (Ortho, IPA, Rom, Cipher) positioned based on shared character set, shared token set, and shared phonology.

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—sharing phonology and tokens—while ciphered text only shares the character set and lacks any additional shared representations. Throughout the paper, we evaluate our hypothesis by analyzing NER performance on seen and unseen languages and analyze in terms of vocabulary 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.

	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.

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

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 named entity recognition. More details on preprocessing and dataset statistics can be found in Appendix A.7. In order to train the model with different input types, we converted all datasets into each input type.

[1]Utilizing https://github.com/antonisa/lang2vec
[2]Words are segmented by white spaces.
[3]https://huggingface.co/datasets/wikimedia/wikipedia

Test Languages	Trained Lang. Set	Ortho	IPA	Rom	Cipher
Seen	sim-same	**0.8466**	0.8085	0.8395	0.8173
	sim-div	0.8409	0.8239	**0.8451**	0.8270
	dissim-same	0.7860	0.7732	**0.7981**	0.7725
	dissim-div	0.7402	0.7524	**0.7538**	0.7518
Unseen	sim-same	0.6611	0.6801	**0.7267**	0.6824
	sim-div	0.6321	0.6787	**0.7151**	0.6772
	dissim-same	0.6626	0.7468	0.7280	**0.7547**
	dissim-div	0.7450	0.7524	**0.7832**	0.7496

Table 2: Average F1 scores for each case. **Bold**: best performing input. Underlined: second best.

4.3 Model Training

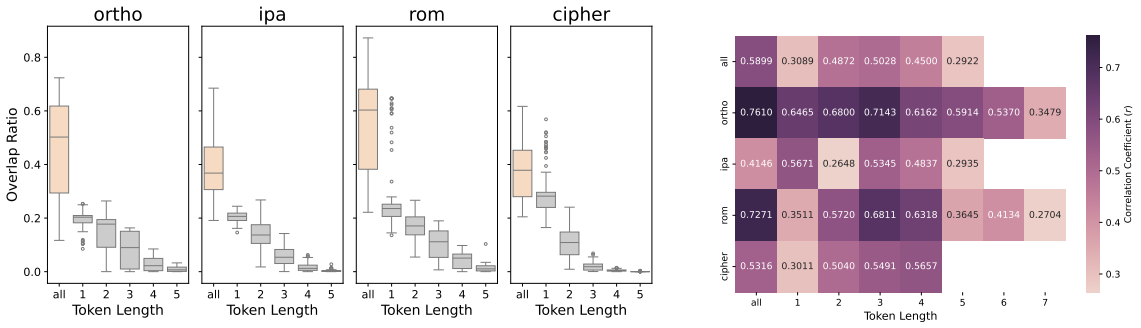
To investigate the impact of different input types, we pre-trained 16 models using four input types and four language sets. We first trained a SentencePiece (character-level) BPE tokenizer for each model with fixed vocabulary size of 30K for all tokenizers. We employed a Transformer architecture, following 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 NER 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: NER Performance across Input Types

Table 2 presents the average F1 scores of each model for seen and unseen languages. p-values obtained from paired t-tests on F1 scores across different input types can be found in Appendix A.5.

Performance within Seen 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 is not statistically significant ($p > 0.05$). On the other hand, Ortho and Rom significantly outperform the other two input types for seen languages ($p < 0.05$).

Performance on Unseen Languages. 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



(a) Overlap ratio distribution.

(b) Correlation between overlap ratio and NER score.

Figure 2: (a) Distribution of lexical overlap ratios across token lengths for different input types. (b) Pearson r between overlap ratios of each token length and NER performance. Correlations with $p > 0.05$ are masked out.

F1 scores in 6 out of 8 cases. Interestingly, contrary to expectation, IPA and Cipher do not show statistically significant differences. We further investigate how Cipher achieves comparable performance, in the following sections.

6 Analysis: Vocabulary Overlap

Transliteration is widely assumed to enhance multilingual language modeling by increasing vocabulary overlap. However, it remains unclear whether conflicting tokens—tokens that are shared but do not form meaningful units (e.g., individual characters)—also contribute to performance. To examine this, we measure lexical overlap of an unseen target language l_t as follows:

$$\text{Lexical Overlap}(l_t) = \max_{l_s \in L_s} \frac{|T_{l_s} \cap T_{l_t}|}{|T_{l_t}|} \quad (1)$$

where l_t is a target language, l_s is one of the pretrained languages L_s, and T_l is set of subword tokens of a dataset in language l.

Transliteration and Lexical Overlap. Figure 2a shows the spread of lexical overlap across token lengths. Ortho and Rom exhibit relatively high overlap across all token lengths, whereas IPA and Cipher show less. Notably, Ciphered text primarily shares single characters across languages rather than longer sequences, reflecting its shared character set without meaningful token overlap. IPA shows relatively high overlap at token length of 2, likely because IPA symbols often form phonemes as character pairs. Meanwhile, Rom and Ortho tend to share longer tokens (length 2–3) across languages, suggesting greater overlap in meaningful subword units.

Vocabulary Overlap and Transferability. To understand how Cipher achieves comparable results on unseen languages, we further investigate how vocabulary overlap associates with task performance. Figure 2b 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 meaningful tokens. To summarize, sharing character tokens does positively correlate with the performance, but having longer tokens in common correlates stronger with the performance.

7 Discussion

Different Patterns on Seen/Unseen Languages. For seen languages, we find that IPA and Cipher lag behind Ortho and Rom. We assume that this is because Ortho and Rom are more likely to share tokens across languages, whereas IPA contains more language-specific symbols and Cipher has little chance of sharing similar character sequences across languages.

Comparable Performance of Cipher. Cipher's comparability to IPA, despite having few shared tokens, highlights the role of a shared character set in transliteration. As IPA symbol sets are inherently language-specific, unseen languages are more likely to produce unknown tokens(`[UNK]`), failing to tokenize appropriately. In contrast, Cipher produces almost no unknown tokens, although the tokens tend to be over-segmented or are segmented in an incoherent manner. This suggests that having conflicting or over-segmented tokens may not be as detrimental as expected in multilingual adaptation scenarios.

8 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. Second, the extrinsic evaluation was limited to a single task—named entity recognition—and it is not immediately obvious that representations that work well for NER would generalize to other tasks (like machine translation, summarization, and question answering). Finally, 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.

9 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.

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)[4] to compute language similarity. Specifically, we extracted syntactic, geographic, and genetic features from lang2vec and computed cosine similarities, denoted as s_{syn}, s_{geo},

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

and s_{gen} in Eq. 2. We also defined lexical similarity s_{lex}, which is obtained by calculating the word overlap ratio between training corpora of each language[5]. Finally, we aggregated all similarity scores (i.e., syntactic, geographic, genetic, and lexical) to derive the overall similarity score between two languages:

$$\begin{aligned} \text{sim}_s(x, y) = \text{s}_{syn}(x, y) + \text{s}_{geo}(x, y) \\ + \text{s}_{gen}(x, y) + \text{s}_{lex}(x, y). \end{aligned} \tag{2}$$

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 $s \in \{\text{sim-same}, \text{sim-div}\}$:

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

.

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

$$\begin{aligned} L_d^* = \arg \min_{\substack{L_d \subset L \\ |L_d|=8}} \Bigg(\frac{1}{|L_d|(|L_d|-1)} \sum_{x \in L_d} \sum_{\substack{y \in L_d \\ y \neq x}} \text{sim}_s(x, y) \\ + \alpha \cdot \Big(\mathbb{1}_{d \in \{\text{sim-div}\}} |\text{SC}_{L_d}| \\ - \mathbb{1}_{d \in \{\text{dissim-div}\}} |\text{SC}_{L_d}| \Big) \Bigg). \end{aligned} \tag{4}$$

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.

[5]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 3 displays the corresponding P-values derived from these tests.

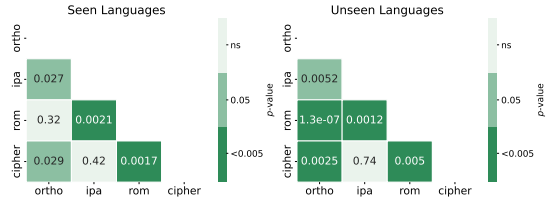


Figure 3: P-value for paired t-test on NER scores across different input types.

A.6 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.7 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[6].

We limited each language with its number of words around 10M[7]. 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. 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.8 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 pretraining and fine-tuning, "Unseen" to those entirely absent during training, and "Zero-Shot" to languages evaluated without task-specific fine-tuning. The language sets differ in terms of typological similarity and script usage. Detailed results for each setting are provided in the respective tables.

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

[7]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.

Lang	Dataset	# Train	# Validate	# Test	Lang	Dataset	# Train	# Validate	# Test
am	wikipedia wikiann	5328 100	- 100	- 100	my	wikipedia wikiann	34309 100	- 100	- 100
ar	wikipedia wikiann	- 20000	- 10000	- 10000	or	wikipedia wikiann	11018 100	- 100	- 100
bn	wikipedia wikiann	28496 10000	- 1000	- 1000	pl	wikipedia wikiann	- 20000	- 10000	- 10000
ca	wikipedia wikiann	26031 20000	- 10000	- 10000	pt	wikipedia wikiann	26510 20000	- 10000	- 10000
ceb	wikipedia wikiann	22724 100	- 100	- 100	ro	wikipedia wikiann	28890 20000	- 10000	- 10000
de	wikipedia wikiann	30460 20000	- 10000	- 10000	ru	wikipedia wikiann	32636 20000	- 10000	- 10000
es	wikipedia wikiann	25727 20000	- 10000	- 10000	si	wikipedia wikiann	23084 100	- 100	- 100
fi	wikipedia wikiann	36190 20000	- 10000	- 10000	so	wikipedia wikiann	5204 100	- 100	- 100
fr	wikipedia wikiann	25353 20000	- 10000	- 10000	sq	wikipedia wikiann	27406 5000	- 1000	- 1000
hi	wikipedia wikiann	25492 5000	- 1000	- 1000	sr	wikipedia wikiann	29961 20000	- 10000	- 10000
hr	wikipedia wikiann	30764 20000	- 10000	- 10000	sv	wikipedia wikiann	29839 20000	- 10000	- 10000
ilo	wikipedia wikiann	5828 100	- 100	- 100	sw	wikipedia wikiann	25911 1000	- 1000	- 1000
ka	wikipedia wikiann	33713 10000	- 10000	- 10000	te	wikipedia wikiann	28543 1000	- 1000	- 1000
ko	wikipedia wikiann	38885 20000	- 10000	- 10000	th	wikipedia wikiann	76083 20000	- 10000	- 10000
lij	wikipedia wikiann	4002 100	- 100	- 100	ur	wikipedia wikiann	23568 20000	- 1000	- 1000
lt	wikipedia wikiann	32836 10000	- 10000	- 10000	uz	wikipedia wikiann	29833 1000	- 1000	- 1000
lv	wikipedia wikiann	31152 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				Zero-Shot			
		Ortho	IPA	Rom	Cip	Ortho	IPA	Rom	Cip	Ortho	IPA	Rom	Cip
Seen	ca	0.9117	0.8970	0.9005	0.9024	0.8997	0.8725	0.8993	0.8803	0.7730	0.6763	0.7770	0.5945
	es	0.8929	0.8759	0.8802	0.9141	0.8773	0.8584	0.8788	0.8657	0.7704	0.5703	0.8095	0.4956
	fr	0.8779	0.8607	0.8717	0.8693	0.8628	0.8252	0.8639	0.8384	0.7293	0.6184	0.7355	0.5645
	lij	0.3269	0.2927	0.4306	0.2775	0.5064	0.4052	0.4615	0.4082	0.2416	0.1702	0.2500	0.1137
	pt	0.8931	0.8842	0.8891	0.8850	0.8798	0.8605	0.8796	0.8674	0.8792	0.8605	0.8796	0.8674
	ro	0.9143	0.9106	0.9153	0.9141	0.9129	0.8855	0.9103	0.8956	0.6257	0.4303	0.6367	0.375
	sq	0.9052	0.8958	0.9011	0.8981	0.9120	0.8738	0.8979	0.8785	0.6791	0.5891	0.7037	0.5335
	sv	0.9300	0.9238	0.9320	0.9311	0.9215	0.8872	0.9247	0.9046	0.4674	0.4093	0.4970	0.4897
Unseen	am	-	-	-	-	0.2000	0.3089	0.3383	0.3623	0.0000	0.1173	0.1004	0.1434
	bn	-	-	-	-	0.8230	0.8907	0.9081	0.8969	0.0000	0.0719	0.1690	0.1372
	de	-	-	-	-	0.8204	0.7400	0.8236	0.7676	0.3323	0.1559	0.4247	0.1307
	fi	-	-	-	-	0.8573	0.8050	0.8609	0.8237	0.4664	0.1993	0.5341	0.1966
	hi	-	-	-	-	0.7395	0.8043	0.8225	0.7861	0.0060	0.1789	0.1343	0.1566
	hr	-	-	-	-	0.8682	0.8318	0.8727	0.8403	0.8682	0.2494	0.5225	0.1794
	ilo	-	-	-	-	0.6400	0.5714	0.6757	0.4498	0.5408	0.3108	0.5408	0.1516
	ka	-	-	-	-	0.6878	0.7920	0.8227	0.7780	0.1160	0.1846	0.1713	0.1126
	ko	-	-	-	-	0.5329	0.7578	0.7883	0.7626	0.0240	0.1483	0.1233	0.0904
	lv	-	-	-	-	0.8940	0.8463	0.8919	0.8695	0.3637	0.2201	0.4402	0.1556
	my	-	-	-	-	0.2286	0.2541	0.2857	0.2232	0.0000	0.0919	0.1474	0.0929
	or	-	-	-	-	0.2738	0.2647	0.3492	0.3533	0.0000	0.0534	0.0000	0.0125
	ru	-	-	-	-	0.8083	0.7842	0.8268	0.8010	0.0894	0.1580	0.2857	0.1273
	sn	-	-	-	-	-	-	-	-	-	-	-	-
	so	-	-	-	-	0.6256	0.4641	0.5500	0.4397	0.3543	0.1662	0.3460	0.2000
	sr	-	-	-	-	0.8574	0.8442	0.8879	0.8691	0.0521	0.1608	0.3454	0.1246
	sw	-	-	-	-	0.8250	0.7381	0.8195	0.7494	0.4311	0.2118	0.4551	0.1765
	te	-	-	-	-	0.3384	0.5336	0.5797	0.5252	0.0033	0.0966	0.0726	0.0550
	th	-	-	-	-	0.4762	0.6637	0.6622	0.6477	0.0009	0.0015	0.0018	0.0059
	ur	-	-	-	-	0.9032	0.9101	0.9273	0.9172	0.0019	0.0253	0.0360	0.0926
	uz	-	-	-	-	0.8266	0.7962	0.8402	0.7862	0.4932	0.2941	0.5120	0.0812

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				Zero-Shot			
		Ortho	IPA	Rom	Cip	Ortho	IPA	Rom	Cip	Ortho	IPA	Rom	Cip
Seen	bn	0.9584	0.9543	0.9524	0.9506	0.9380	0.9375	0.9466	0.9377	0.9380	0.9375	0.9466	0.9377
	fr	0.8779	0.8607	0.8717	0.8693	0.8436	0.8255	0.8430	0.8378	0.3384	0.3728	0.3891	0.3826
	hi	0.8909	0.8695	0.8890	0.8877	0.8524	0.8577	0.8394	0.8314	0.5151	0.5167	0.5400	0.4647
	hr	0.8986	0.8876	0.8931	0.8950	0.8741	0.8527	0.8767	0.8605	0.3666	0.3353	0.3952	0.3762
	or	0.6032	0.6584	0.6235	0.6721	0.5483	0.4981	0.5873	0.4962	0.1338	0.2266	0.2731	0.1948
	ru	0.8614	0.8515	0.8604	0.8578	0.8395	0.8286	0.8375	0.8304	0.2268	0.2365	0.2369	0.2348
	sr	0.9099	0.8413	0.9175	0.9117	0.8918	0.8484	0.8969	0.8900	0.2257	0.1228	0.3048	0.2571
	ur	0.9447	0.9410	0.9476	0.9408	0.9396	0.9424	0.9333	0.9318	0.2834	0.2518	0.3534	0.2200
Unseen	am	-	-	-	-	0.0079	0.2902	0.3282	0.2695	0.000	0.1063	0.1017	0.0155
	de	-	-	-	-	0.7934	0.7381	0.8047	0.7608	0.1986	0.1257	0.2420	0.1048
	es	-	-	-	-	0.8511	0.8144	0.8573	0.8265	0.2640	0.1913	0.3043	0.1641
	fi	-	-	-	-	0.8427	0.7993	0.8460	0.8201	0.2539	0.1717	0.2688	0.1820
	ilo	-	-	-	-	0.5333	0.5356	0.5537	0.4627	0.2473	0.2500	0.2922	0.1313
	ka	-	-	-	-	0.5860	0.7961	0.8162	0.7872	0.0181	0.1207	0.1851	0.1314
	ko	-	-	-	-	0.5244	0.7318	0.7792	0.7577	0.0026	0.1445	0.1538	0.1262
	lij	-	-	-	-	0.3071	0.3684	0.2975	0.3064	0.1183	0.1037	0.2172	0.1022
	lv	-	-	-	-	0.8826	0.8468	0.8891	0.8605	0.1710	0.1945	0.3140	0.0941
	my	-	-	-	-	0.1596	0.1721	0.2975	0.2424	0.0000	0.0263	0.0912	0.0552
	pt	-	-	-	-	0.8535	0.8206	0.8547	0.8312	0.2351	0.1104	0.3257	0.0947
	ro	-	-	-	-	0.8889	0.8754	0.8963	0.8695	0.2116	0.1710	0.2407	0.0537
	sn	-	-	-	-	-	-	-	-	-	-	-	-
	so	-	-	-	-	0.4874	0.4870	0.5128	0.5236	0.3175	0.1930	0.2759	0.1554
	sq		-	-	-	0.8557	0.8319	0.8604	0.8315	0.3258	0.2241	0.3604	0.0944
	sv	-	-	-	-	0.9059	0.8583	0.9043	0.8850	0.1923	0.0546	0.1745	0.0810
	sw	-	-	-	-	0.7634	0.7429	0.7955	0.7359	0.1996	0.0843	0.2402	0.0893
	te	-	-	-	-	0.3297	0.5753	0.6440	0.5119	0.0000	0.1277	0.1955	0.0514
	th	-	-	-	-	0.3531	0.6680	0.6479	0.6302	0.0001	0.0049	0.0025	0.0032
	uz	-	-	-	-	0.8384	0.7819	0.8360	0.7863	0.1377	0.0588	0.1310	0.0130

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				Zero-Shot			
		Ortho	IPA	Rom	Cip	Ortho	IPA	Rom	Cip	Ortho	IPA	Rom	Cip
Seen	de	0.8716	0.8518	0.8599	0.8622	0.8184	0.7924	0.8248	0.8095	0.3300	0.2076	0.3288	0.2098
	fi	0.8855	0.8813	0.8850	0.8861	0.8618	0.8264	0.8638	0.8436	0.3129	0.1859	0.3216	0.2062
	ilo	0.6053	0.6216	0.6881	0.6996	0.6757	0.7123	0.6368	0.6549	0.4615	0.1965	0.3170	0.2677
	lv	0.9284	0.9205	0.9232	0.9230	0.8995	0.8736	0.9006	0.8998	0.2784	0.1720	0.2608	0.1711
	sn	-	-	-	-	-	-	-	-	-	-	-	-
	so	0.6111	0.5648	0.6000	0.5249	0.5551	0.5887	0.6577	0.5556	0.3891	0.3128	0.3291	0.2731
	sw	0.8481	0.8385	0.8532	0.8481	0.8291	0.7981	0.8421	0.8125	0.3126	0.1655	0.2427	0.1624
	uz	0.8648	0.8655	0.8665	0.8836	0.8621	0.8210	0.8608	0.8314	0.8621	0.8210	0.8608	0.8314
Unseen	am	-	-	-	-	0.2833	0.5560	0.2845	0.5018	0.0402	0.0730	0.0429	0.0121
	bn	-	-	-	-	0.8269	0.8791	0.9005	0.9430	0.0415	0.1208	0.0697	0.0234
	ca	-	-	-	-	0.8733	0.8255	0.8750	0.8542	0.2548	0.1487	0.2590	0.0972
	es	-	-	-	-	0.8518	0.8103	0.8583	0.8377	0.2646	0.1242	0.2846	0.0874
	fr	-	-	-	-	0.8312	0.7607	0.8294	0.8447	0.2633	0.1434	0.2878	0.0946
	hi	-	-	-	-	0.7128	0.8210	0.8055	0.7981	0.0030	0.0890	0.0910	0.0945
	hr	-	-	-	-	0.8531	0.8404	0.8532	0.8495	0.2098	0.0930	0.2079	0.0604
	ka	-	-	-	-	0.6289	0.8577	0.8103	0.8606	0.0591	0.0724	0.1008	0.0488
	ko	-	-	-	-	0.5282	0.8297	0.7652	0.8381	0.0590	0.0666	0.0559	0.0484
	lij	-	-	-	-	0.3319	0.2893	0.3333	0.2979	0.1393	0.0836	0.1337	0.0663
	my	-	-	-	-	0.2128	0.5263	0.2785	0.5750	0.0000	0.0214	0.0566	0.0310
	or	-	-	-	-	0.0708	0.4082	0.3851	0.2339	0.0090	0.0137	0.1102	0.1208
	pt	-	-	-	-	0.8566	0.8015	0.8558	0.8449	0.2773	0.1053	0.2992	0.0836
	ro	-	-	-	-	0.8906	0.8548	0.8880	0.8768	0.1894	0.1050	0.2112	0.0694
	ru	-	-	-	-	0.7992	0.7922	0.8132	0.8051	0.0377	0.0880	0.1506	0.0762
	sq	-	-	-	-	0.8658	0.8120	0.8627	0.8259	0.2117	0.1075	0.2447	0.0785
	sr	-	-	-	-	0.8540	0.8201	0.8790	0.8739	0.0243	0.1693	0.2781	0.1292
	sv	-	-	-	-	0.9075	0.8484	0.9076	0.8919	0.2286	0.0546	0.2230	0.0520
	te	-	-	-	-	0.3278	0.7441	0.5494	0.7632	0.0100	0.0619	0.0309	0.0373
	th	-	-	-	-	0.5162	0.6841	0.6320	0.6110	0.0021	0.0074	0.0089	0.0232
	ur	-	-	-	-	0.8906	0.9208	0.9205	0.9220	0.0419	0.0138	0.0757	0.0762

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	am	0.4796	0.4615	0.5388	0.5203	0.4941	0.5403	0.5760	0.5364
	bn	0.9584	0.9100	0.9524	0.9506	0.9579	0.9488	0.9552	0.9479
	fr	0.8779	0.8607	0.8717	0.8693	0.8528	0.8265	0.8487	0.8432
	ka	0.8866	0.8873	0.8850	0.8837	0.8647	0.8607	0.8619	0.8598
	ko	0.8611	0.8576	0.8623	0.8628	0.7699	0.8347	0.8382	0.8333
	my	0.5401	0.5852	0.5617	0.5188	0.5259	0.5738	0.5164	0.5477
	te	0.7880	0.7983	0.7822	0.7922	0.7532	0.7529	0.7528	0.7734
	th	0.7052	0.6880	0.6656	0.6726	0.7031	0.6813	0.6810	0.6727
Unseen	ca	-	-	-	-	0.8797	0.8503	0.8803	0.8513
	de	-	-	-	-	0.8088	0.7555	0.8134	0.7855
	es	-	-	-	-	0.8615	0.8315	0.8687	0.8352
	fi	-	-	-	-	0.8504	0.8188	0.8532	0.8311
	hi	-	-	-	-	0.6585	0.8223	0.8472	0.7939
	hr	-	-	-	-	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
	lv	-	-	-	-	0.8948	0.8544	0.891	0.8762
	or	-	-	-	-	0.3840	0.3931	0.4373	0.2913
	pt	-	-	-	-	0.8609	0.8245	0.8630	0.8437
	ro	-	-	-	-	0.8940	0.8746	0.8979	0.8788
	ru	-	-	-	-	0.6753	0.7941	0.8207	0.8049
	sn	-	-	-	-	-	-	-	-
	so	-	-	-	-	0.6140	0.4893	0.5462	0.5299
	sq	-	-	-	-	0.8720	0.8395	0.8533	0.8389
	sr	-	-	-	-	0.6697	0.8405	0.8826	0.8735
	sv	-	-	-	-	0.9117	0.8641	0.9119	0.8920
	sw	-	-	-	-	0.7968	0.7527	0.7855	0.7536
	ur	-	-	-	-	0.6974	0.9072	0.9243	0.9226
	uz	-	-	-	-	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)