# Phoneme Bridges: Leveraging Phonetic Similarity for Low-Resource Language Understanding

**Chris Ge** and **Brian Le** and **Daria Kryvosheieva**
Massachusetts Institute of Technology
cge7@mit.edu and brianle@mit.edu and daria_k@mit.edu

## Abstract

Currently, the majority of the world's languages are "low-resource," meaning that they have little digital training data and thus lack adequate NLP technologies. We propose a method to improve language models' understanding of low-resource languages without extensive training by leveraging information from related high-resource languages. Our method is based on augmenting the original input with phonetic (romanization) input, making it especially useful for languages that are similar in spoken speech but use different, non-Latin writing systems. We select Hindi—Urdu as an example of such a language pair and investigate whether fine-tuning on partially romanized Hindi datasets improves performance in Urdu for a variety of NLP tasks. We find that our method produces improvements, but they are not statistically significant because the model does not reliably learn to exploit romanizations for the tasks. [1]

## 1 Introduction

At present, natural language processing (NLP) technologies are far from being universally accessible due to not being sufficiently linguistically diverse. According to Joshi et al. (2020), 90% of languages, which include over 1 billion speakers in total, are ignored by modern language technologies. The cause of the lack of diversity lies in the severe disparity in the availability of digital resources across languages. For instance, on Wikipedia, the most well-represented language (English) contains nearly 7 million articles, while the median languages by representation (Tamazight and Zulu) contain only around 10,000 articles (Wikipedia, 2024).

Attempts have been made to address this disparity between languages by pretraining multilingual large language models such as mBERT (Devlin et al., 2019) using multiple **source languages**. While these multilingual pretrained language models (PLMs) do exhibit improved cross-lingual performance when the **target language** is a high-resource language (HRL) (Conneau et al., 2020), they still perform significantly worse when the target language is a low-resource language (LRL) (Wu and Dredze, 2020). This difference in cross-lingual performance is partially because PLMs perform better when languages share similar scripts, which is the case in many HRLs but few LRLs (Muller et al., 2021).

For LRLs with rarer scripts, we can instead exploit shared pronunciation to improve cross-lingual transfer. For example, languages may borrow words from each other, keeping their pronunciations but transcribing them using their own writing systems. These words differ orthographically but have the same meaning that we can draw upon using their similar sounds (Nguyen et al., 2024). Previous studies have found that using a source and target language pair with a common language family and many similar-sounding words is correlated with better cross-lingual transfer (Fujinuma et al., 2022; Nguyen et al., 2024).

Hindi: **क्या आप ठीक हैं?** (Kyaa aap theek hain?)
Urdu: **کیا آپ ٹھیک ہیں؟** (Kyaa aap theek hain?)
*"Are you okay?"*

Figure 1: We choose Hindi and Urdu as an example pair of languages with different scripts but a largely shared lexicon. Here, the sentences written in Hindi and in Urdu look completely different, but romanization reveals their shared meaning.

We aim to improve a PLM's downstream performance on NLP tasks in an LRL by a series of fine-tuning steps that exploit pronunciation similarities between an HRL and a closely related LRL. We use Hindi as the HRL and Urdu as the LRL, as these two Indic languages share 70% of their lexicon

[1] Our code and data are publicly available at https://github.com/brian224code/phoneme-bridges.

but utilize different writing systems, allowing us to exploit the lexical similarities for transfer learning. We integrate phonetic information by using a transliterator to obtain the romanized version of the training text and concatenating it with the original text. Phonemic transcriptions would better capture the underlying similarity between languages, but obtaining accurate phonemic transcriptions often requires a language-specific transformer-based grapheme-to-phoneme model, which may not be available for low-resource languages. General tools for grapheme-to-phoneme conversion like Epitran (Mortensen et al., 2018) are often incomplete: Epitran, for instance, omits short vowels in its Urdu phoneme transcriptions. Romanization reflects surface forms of words (physically produced while speaking) instead of underlying forms (those in the speakers' mind), which may slightly obscure similarities, but it is widely available for all languages through general tools like UROMAN and language-family-specific romanizers (Purkayastha et al., 2023). Since romanization tools are accessible for all languages, obtaining phonetic information via romanization aligns better with our goal of addressing PLM disparity in LRLs.

## 2 Related Work

### Cross-Lingual Transfer Using Phonemic or Phonetic Information

Recent works have successfully integrated phonemic information with orthographic information by designing a model architecture that includes a phonemic embedding layer and training the model on a large dataset of mixed graphemes and phonemes with a loss function that combines the two modalities; this includes models like PhoneXL (Nguyen et al., 2023, 2024), PhonemeBERT (Sundararaman et al., 2021), XPhoneBERT (The Nguyen et al., 2023), and Mixed-Phoneme BERT (Zhang et al., 2022). These works all find improvements in their models' performance on token-level and/or natural language understanding tasks in the target LRL over a multilingual PLM like mBERT or XLM-R (Conneau et al., 2020). However, because of the large amount of training required to learn the full set of parameters associated with the phoneme representation, these approaches are computationally expensive, making it difficult to bring new developments in state-of-the-art large language models to use in LRLs.

Another approach is to incorporate phonetic information by continuing to train a PLM instead of designing a new model architecture. An example of this approach is RomanSetu (Jaavid et al., 2024), which starts with a pretrained LLaMA-2-7B model (Touvron et al., 2023), continues pretraining on 500 million words in the target LRL transliterated into romanization, and then instruction-tunes on romanized tasks in the LRL. Although this approach doesn't learn phonetic representations from scratch, even this amount of unlabeled extra training data in the LRL is unreasonable to expect: Joshi et al. (2020) categorized 88% of languages as having little to no unlabeled data online, so that even unsupervised methods wouldn't help them. Purkayastha et al. (2023) reduce the training requirements further by only fine-tuning learning adapters for an mBERT model on romanized LRL datasets. They find improved performance compared to fine-tuning on non-transliterated data, especially on languages written in scripts not included in the model's training data. Their results support the idea that fine-tuning on romanization rather than regular text can better leverage the PLM's existing source language understanding, but their analysis focuses more on comparing transliterators than maximizing the effectiveness of the learned romanization representation for downstream performance. We will exploit the fact that some LRLs have similar phonetic properties and thus romanizations to an HRL through our supplementary fine-tuning step that first learns romanization representations in the HRL.

### Supplementary Training on Intermediate Labeled-data Tasks (STILTs)

While previous works required large amounts of data to incorporate phonetic information into PLMs, we desire a method that achieves comparable performance using minimal LRL data. STILTs is an easy-to-implement method to improve a PLM's general performance on downstream tasks, even after pretraining is complete (Phang et al., 2018). In the STILTs training method, the PLM is first fine-tuned on a data-rich, supervised intermediate task (the **supplementary** task), and then further fine-tuned on a downstream task of interest (the **target** task), improving performance and decreasing variance on the target task. The authors also observed that STILTs is particularly effective when the target task is data-poor, and when the intermediate task is closely related to the target task, both of which apply to our HRL and LRL train-

ing situation. Phang et al. (2020) further showed that English STILTs can specifically improve cross-lingual transfer.

The STILTs method has also been used to assist a model in learning phoneme representations. BORT (Gale et al., 2023) demonstrated that intermediately fine-tuning BART by replacing words with their International Phonetic Alphabet (IPA) spellings and leading the model to reconstruct the sentence can be used to learn phonemic representations. However, since a pre-trained BART is only available for English, their work can't be directly used for cross-lingual transfer.

To summarize, fine-tuning on romanized versions of an LRL dataset, using STILTs for English, and using a closely related source language have been separately found to improve cross-lingual transfer, and STILTs can also be used to learn phonemic representations. As far as we know, **we are the first to use romanization-augmented STILTs in the HRL to transfer learned phonetic and semantic knowledge to a romanization-augmented LRL task.** Our training method would reduce the need for data in LRLs by transferring learning from phonetically similar HRLs, lowering the barrier to achieving performant models in LRLs.

## 3 Methods

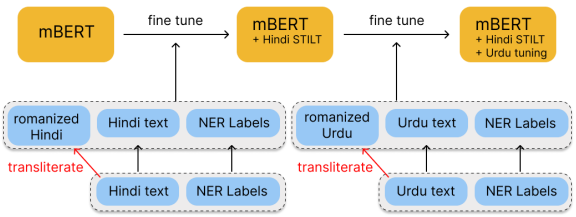



Figure 2: Our training pipeline for the task of NER.

We will first give an overview of our training pipeline for integrating romanization information from the HRL and the LRL, and then specify our implementation details for each step of the pipeline.

1. Pick an NLP task. We'll experiment with named entity recognition (NER) and part-of-speech (POS) tagging.

2. Gather a dataset for the task in each of the LRL and the HRL. Retrieve the romanizations of the two datasets' input texts using a transliterator.

3. Fine-tune the PLM on the NLP task in the HRL, randomly replacing a fixed proportion of words in the input text of the data by their romanizations.

4. Further fine-tune and evaluate the resulting model on the LRL task with both text and romanization.

### 3.1 Datasets

We use the Google Xtreme benchmark (Hu et al., 2020), which was created to evaluate how well cross-lingual learning methods transfer linguistic knowledge. The benchmark contains multiple tasks in both the HRL and the LRL of our choice: Hindi and Urdu. We specifically use the PAN-X and UD-POS datasets for both languages, which cover NER and POS tagging tasks, respectively.

### 3.2 Romanization Scheme

We use the ai4bharat romanization tool (Kunchukuttan et al., 2020) to obtain the romanizations of Hindi and Urdu words in our datasets. We chose ai4bharat because it exhibited better downstream performance on token-level tasks than UROMAN in Purkayastha et al. (2023). Even though ai4bharat is only available for Indic languages, our work is primarily to explore if the concept of exploiting phonetic simliarities through romanization is possible, and in practice the same methodology can be applied to other languages using UROMAN.

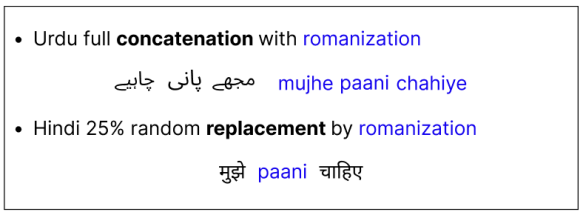



Figure 3: An example illustrating our romanization scheme. We concatenate the romanizations in the LRL and replace some text with romanizations in the HRL.

After obtaining the romanizations for the LRL datasets, we **concatenate** the romanizations with the original text, and for the HRL datasets, we randomly **replace** 25% of the original words with their romanization equivalents. In the LRL case, we concatenate the romanizations instead of replacing the original script entirely because Nguyen et al. (2024) found that romanization is a complementary

signal but not a replacement for orthographic representation in terms of improving task performance. However, for the HRL intermediate fine-tuning, our goal is not to maximize performance on the test set, but to learn useful representations for romanization to transfer to the LRL. To this end, we hope that this random replacement will encourage the model not to ignore the romanized words, while also providing Hindi context that's already present in its training data. As for the choice of 25%, in BORT (Gale et al., 2023), phonetic replacements were performed at a rate of 10%, but replacement rates as high as 35% have been used (Liu et al., 2020). We chose 25% as an intermediate value.

### 3.3 Trained Models

We chose multilingual BERT (mBERT, Devlin et al., 2019) as this multilingual PLM is commonly used for testing cross-lingual transfer (Pfeiffer et al., 2020; Muller et al., 2021; Purkayastha et al., 2023) and using it will allow us to determine if our methods improve the performance of readily available multilingual PLMs. For each of the PAN-X and UD-POS datasets, we fine-tune the following four versions of mBERT on one A100 GPU on the OpenMind Computing Cluster at MIT.

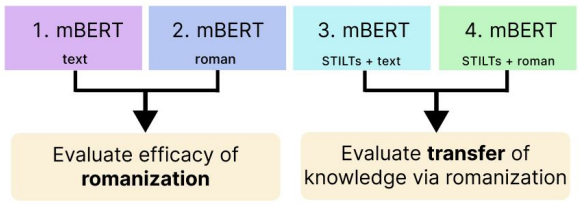



Figure 4: Our four fine-tuned models and relevant comparisons.

1. **mBERT$_{\text{text}}$**: mBERT fine-tuned directly on the Urdu dataset without romanizations.
2. **mBERT$_{\text{roman}}$**: mBERT fine-tuned directly on the Urdu dataset with romanizations concatenated.
3. **mBERT$_{\text{STILTs + text}}$**: mBERT intermediately fine-tuned on the Hindi dataset, then further fine-tuned on the Urdu dataset, all without romanizations.
4. **mBERT$_{\text{STILTs + roman}}$**: mBERT intermediately fine-tuned on the Hindi dataset with 25% of the words replaced with romanizations, then further fine-tuned on the Urdu dataset with romanizations concatenated.

We will compare mBERT$_{\text{text}}$ to mBERT$_{\text{roman}}$ and mBERT$_{\text{STILTs + text}}$ to mBERT$_{\text{STILTs + roman}}$ by their performances on the test split of the dataset, as depicted in Figure 4.

The first comparison will demonstrate the efficacy of directly augmenting Urdu data with romanization. It will give us a measure of how much of the performance improvement from adding romanization to our training processes is due to the PLM already having the ability to utilize romanization. This hypothetical ability may stem from the PLM's bias for Latin script languages and the romanization already in its training data.

The second comparison is the more important one. It measures if adding the Hindi and Urdu romanizations improves on cross-lingual transfer performance over just having a plain Hindi fine-tune. This is what we're interested in: can the model use romanizations as a bridge to transfer knowledge between Hindi and Urdu about the task?

### 3.4 Evaluation Metrics

For PAN-X (NER), we will measure the **macro-F1** score using `sklearn.metrics.f1_score` with `average='macro'`. The macro-F1 score represents the average F1 score (harmonic mean of precision and recall) over all class labels. Given that the NER task is framed as a token classification task (determining for each input token which type of named entity it is a part of), macro-F1 score is a very intuitive metric to use, and it has been used to evaluate NER performance in the literature (Abilio et al., 2024).

For UD-POS (POS tagging), we will also report **macro-F1** score because it is another token classification task (determining the part of speech for each input token).

## 4 Results

| Model | POS Tagging Score | NER Score |
|---|---|---|
| mBERT$_{\text{text}}$ | 0.8700 | 0.9770 |
| mBERT$_{\text{roman}}$ | **0.8728** | **0.9780** |
| mBERT$_{\text{STILTs+text}}$ | 0.8702 | 0.9763 |
| mBERT$_{\text{STILTs+roman}}$ | **0.8735** | **0.9788** |

Table 1: Our results (macro-F1 scores) for POS tagging and NER. The better F1 score in each comparison between two models is bolded.

We trained each of the four models described in section 3.3 on both the POS tagging and NER

| Comparison | Task | Mean Difference | 95% Confidence Interval | P-value |
|---|---|---|---|---|
| $\text{mBERT}_{\text{roman}} - \text{mBERT}_{\text{text}}$ | POS | 0.0029 | [-0.0016, 0.0071] | 0.2180 |
| | NER | 0.0010 | [-0.0066, 0.0079] | 0.7680 |
| $\text{mBERT}_{\text{STILTs+roman}} - \text{mBERT}_{\text{STILTs+text}}$ | POS | 0.0035 | [-0.0019, 0.0092] | 0.2200 |
| | NER | 0.0026 | [-0.0038, 0.0095] | 0.4100 |

Table 2: Statistical test results comparing model macro-F1 scores. We observed an increase in all macro-F1 scores for models with romanization, but the differences were not statistically significant at the 5% significance threshold.

datasets, and evaluated the 8 resulting models. Table 1 shows the models' macro-F1 scores on the testing dataset.

To analyze the statistical significance of the difference between the romanized and non-romanized models in each comparison, we took 10000 bootstrapped resamplings of our evaluation data and used them to calculate both a 95% confidence interval for the difference in macro-F1 scores, as well as the p-value of a two-tailed test. Table 2 shows the results: while our hypothesis was correct in that adding romanizations led to improvement and the combination of Hindi intermediate fine-tuning and romanizations produced the highest scores overall, the improvements were not statistically significant.

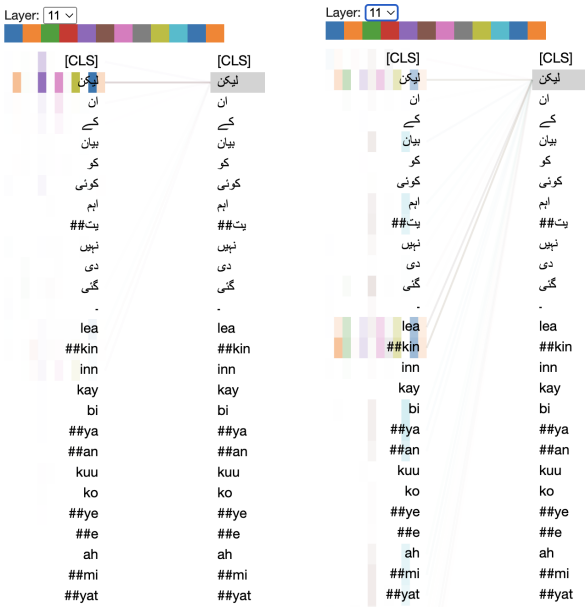
## 5 Discussion

In accordance with the comparison plan from Figure 4, we analyze the contribution of romanizations to both the Urdu task itself and the cross-lingual transfer performance.

The lack of statistically significant improvement from $\text{mBERT}_{\text{text}}$ to $\text{mBERT}_{\text{roman}}$ means that augmenting with romanization information alone doesn't help with learning the task in Urdu; the PLM is unable to directly make effective use of romanizations beyond what it already gains from the text. This is not too surprising, as romanized Urdu text likely does not make up a significant portion of the PLM's pretraining data.

The lack of significant improvement from $\text{mBERT}_{\text{STILTs + text}}$ to $\text{mBERT}_{\text{STILTs + roman}}$ means adding romanization through our pipeline does not significantly improve cross-lingual transfer performance. This is the main negative result for our hypothesis that our training pipeline would improve cross-lingual transfer. It is possible that to effectively use romanizations for the task, either a larger amount of training data or more targeted adapters for cross-lingual transfer are necessary, as was the case in the successful related works.

Despite the negative result, our training pipeline still showed potential for learning Urdu romanizations based on Hindi training data. Using the BertViz (Vig, 2019) tool for visualizing attention, we inspected the final layers of both the $\text{mBERT}_{\text{STILTs+roman}}$ and $\text{mBERT}_{\text{roman}}$ models fine-tuned on UD-POS, and found that $\text{mBERT}_{\text{STILTs+roman}}$ (fine-tuned on Hindi data) paid significant attention to the Urdu romanizations of the words, while $\text{mBERT}_{\text{roman}}$ did not, as seen in Figure 5. This implies that the additional **Hindi** fine-tune helped the model learn to pay attention to **Urdu** romanizations, supporting our hypothesis that the model can transfer learning from Hindi romanizations to Urdu romanizations due to their simliarity. However, for the models fine-tuned on PAN-X, both models paid attention to the romanization, even after controlling the sizes of the Hindi and Urdu datasets to be the same as in UD-POS.




(a) $\text{mBERT}_{\text{roman}}$ (b) $\text{mBERT}_{\text{STILTs+roman}}$

Figure 5: Comparison of $\text{mBERT}_{\text{roman}}$ and $\text{mBERT}_{\text{STILTs+roman}}$ final layer attention using BertViz. "Lea-kin" is the romanization of the highlighted Urdu token. We see that in $\text{mBERT}_{\text{STILTs+roman}}$, much more attention is paid to the romanization, implying that the STILTs method helped the model learn the relationship between Urdu text and romanization.

We hypothesize that there is some property of the Urdu PAN-X dataset that makes it easier to learn romanization representations from: if the Urdu-only model already learns the romanizations properly, then adding romanized Hindi isn't necessary to help learn Urdu romanizations.

These visualizations all suggest that our main negative result is due to the model's failure to learn how to properly exploit the romanizations for the task, not due to a failure in learning representations for the romanizations themselves. Thus, future work could potentially still use our pipeline in applications that require phonetic representations in an LRL.

## 6 Conclusion

We have evaluated a new process for improving cross-lingual transfer performance between an HRL and a phonetically similar LRL. Our process augments the training data in both the HRL and LRL with romanizations and performs two fine-tuning steps, first in the HRL, then in the LRL. Although our training process did not significantly improve task performance in the LRL, we found that adding the HRL fine-tuning step can cause the model to pay attention to the corresponding romanizations for each word in the LRL when it didn't before. This implies that our process can help the model learn romanization representations in the LRL using data from the HRL, which is potentially useful for applications involving phonetic information in LRLs. Looking forward, we hope to motivate more work focused on learning in related HRL-LRL pairs to improve NLP technologies in the LRL.

## 7 Limitations

In this paper, we specifically explored the cross-lingual transfer abilities of our pipeline using mBERT, with Hindi as the HRL and Urdu as the LRL, on the two NLP tasks of POS tagging and NER. In contrast, our goal was to make a claim about the effects of our pipeline on any language model, on any HRL-LRL pair with sufficient phonetic similarity, and on any NLP task of interest. Due to time and compute restraints, we had to narrow our work down to these specific choices of model, languages, and dataset, which we hope are representative of the general problem.

Future work could also experiment with using phonemic transcriptions instead of romanization to better capture underlying similarity. However, as we noted in our choice to use romanization, phonemic transcriptions are generally more difficult to access than romanization, making them unrealistic for practical use in LRLs.

## 8 Impact Statement

By introducing a method that leverages phonetic similarities between HRLs and related LRLs, we provide a practical and accessible approach to enhance cross-lingual transfer learning. Although our framework did not yield statistically significant improvements, the minor improvements we did see suggest that romanization, and more generally, phonetic information, can potentially be a tool for bridging the performance gap between HRLs and LRLs. Our framework also shows promise in enabling models to utilize phonetic information.

Our work addresses disparities in linguistic diversity within NLP. By focusing on leveraging widely available tools like romanizers, our method aligns with the broader goal of making NLP more inclusive and accessible. Improving the performance of NLP technologies in LRLs is particularly important to ensure the benefits from advancements in the field reach as many people as possible. Future applications of this framework have the potential to improve NLP capabilities for LRLs, supporting linguistic diversity and aiding communities in preserving their languages in the digital age.

## References

Ramon Abilio, Guilherme Palermo Coelho, and Ana Estela Antunes da Silva. 2024. Evaluating named entity recognition: A comparative analysis of mono- and multilingual transformer models on a novel brazilian corporate earnings call transcripts dataset. *Applied Soft Computing*, 166:112158.

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for*

*Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. 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.

Robert C. Gale, Alexandra C. Salem, Gerasimos Fergadiotis, and Steven Bedrick. 2023. Mixed orthographic/phonemic language modeling: Beyond orthographically restricted transformers (BORT). In *Proceedings of the 8th Workshop on Representation Learning for NLP (RepL4NLP 2023)*, pages 212–225, Toronto, Canada. Association for Computational Linguistics.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *International Conference on Machine Learning*, pages 4411–4421. PMLR.

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

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the nlp world. *arXiv preprint arXiv:2004.09095*.

Anoop Kunchukuttan, Divyanshu Kakwani, Satish Golla, Avik Bhattacharyya, Mitesh M Khapra, Pratyush Kumar, et al. 2020. Ai4bharat-indicnlp corpus: Monolingual corpora and word embeddings for indic languages. *arXiv preprint arXiv:2005.00085*.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

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)*.

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.

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.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. Unks everywhere: Adapting multilingual language models to new scripts. *arXiv preprint arXiv:2012.15562*.

Jason Phang, Iacer Calixto, Phu Mon Htut, Yada Pruksachatkun, Haokun Liu, Clara Vania, Katharina Kann, and Samuel R Bowman. 2020. English intermediate-task training improves zero-shot cross-lingual transfer too. *arXiv preprint arXiv:2005.13013*.

Jason Phang, Thibault Févry, and Samuel R Bowman. 2018. Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. *arXiv preprint arXiv:1811.01088*.

Sukannya Purkayastha, Sebastian Ruder, Jonas Pfeiffer, Iryna Gurevych, and Ivan Vulić. 2023. Romanization-based large-scale adaptation of multilingual language models. *arXiv preprint arXiv:2304.08865*.

Mukuntha Narayanan Sundararaman, Ayush Kumar, and Jithendra Vepa. 2021. Phonemebert: Joint language modelling of phoneme sequence and asr transcript. In *Interspeech 2021*, pages 3236–3240.

Linh The Nguyen, Thinh Pham, and Dat Quoc Nguyen. 2023. Xphonebert: A pre-trained multilingual model for phoneme representations for text-to-speech. In *INTERSPEECH 2023*, pages 5506–5510.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Jesse Vig. 2019. A multiscale visualization of attention in the transformer model. *arXiv preprint arXiv:1906.05714*.

Wikipedia. 2024. List of wikipedias.

Shijie Wu and Mark Dredze. 2020. Are all languages created equal in multilingual BERT? In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 120–130, Online. Association for Computational Linguistics.

Guangyan Zhang, Kaitao Song, Xu Tan, Daxin Tan, Yuzi Yan, Yanqing Liu, Gang Wang, Wei Zhou, Tao Qin, Tan Lee, and Sheng Zhao. 2022. Mixed-phoneme bert: Improving bert with mixed phoneme and sup-phoneme representations for text to speech. In *Interspeech 2022*, pages 456–460.