INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

XPhoneBERT: A Pre-trained Multilingual Model for Phoneme Representations for Text-to-Speech

Linh The Nguyen, Thinh Pham, Dat Quoc Nguyen

VinAI Research, Vietnam
{v.linhnt140, v.thinhphp1, v.datnq9}@vinai.io

Abstract

We present XPhoneBERT, the first multilingual model pre-trained to learn phoneme representations for the downstream text-to-speech (TTS) task. Our XPhoneBERT has the same model architecture as BERT-base, trained using the RoBERTa pre-training approach on 330M phoneme-level sentences from nearly 100 languages and locales. Experimental results show that employing XPhoneBERT as an input phoneme encoder significantly boosts the performance of a strong neural TTS model in terms of naturalness and prosody and also helps produce fairly high-quality speech with limited training data. We publicly release our pre-trained XPhoneBERT with the hope that it would facilitate future research and downstream TTS applications for multiple languages.

Index Terms: XPhoneBERT, Multilingual model, Pre-trained model, Phoneme representation, Text-to-speech, Neural TTS, Speech synthesis.

1. Introduction

Advancements in neural TTS technology have led to significant improvements in producing natural-sounding speech [1, 2, 3, 4], increasingly closing the gap between artificial speech and human-recorded speech in terms of naturalness. Early work such as [5] employs an encoder to directly convert input raw texts to mel-spectrograms that are then fed into a decoder to generate output speech. Other works often take phoneme sequences as input for their encoder [6, 7, 8, 9]. Here, the encoder in these works might be extended by utilizing recent large-scale pre-trained language models that are learned from unlabeled textual or phonemic description data to enhance the naturalness of speech outputs.

The large-scale pre-trained language models, e.g. BERT [10], RoBERTa [11] and ALBERT [12], have proved their effectiveness, improving state-of-the-art performances of various natural language processing research and application tasks. For TTS, some works incorporate contextualized word embeddings generated by the pre-trained BERT [10] into their standard encoder [13, 14, 15]. In general, an input phoneme sequence is fed into the standard TTS encoder to produce phoneme representations, while its corresponding raw text is fed into BERT to obtain contextualized word embeddings. To construct the input vectors of the TTS decoder, the produced representations of the input phonemes are concatenated with the BERT-based contextualized embedding of the corresponding word that the phonemes belong to. As a result, BERT helps increase the quality of the output synthesized speech. Here, the pre-trained BERT is used to provide additional contextual information for phoneme representations indirectly. Therefore, it might be better if the contextualized phoneme representations are directly produced by a pre-trained BERT-type model that is learned from unlabeled phoneme-level data.

Recent works confirm that pre-trained models for phoneme representations, including PnG BERT [16], Mixed-Phoneme BERT [17] and Phoneme-level BERT [18], help improve advanced TTS systems. PnG BERT and Mixed-Phoneme BERT are trained based on the BERT pre-training approach [10], in which PnG BERT takes both phonemes and graphemes (i.e. subword tokens) as the input, while Mixed-Phoneme BERT takes both phonemes and sup-phoneme tokens as the input. Phoneme-level BERT is trained based on the ALBERT pre-training approach [12], only taking phonemes as the input. In addition to the standard masked token prediction task as used in PnG BERT and Mixed-Phoneme BERT, the Phoneme-level BERT also proposes an additional auxiliary task that predicts the corresponding grapheme for each phoneme. Here, PnG BERT, Mixed-Phoneme BERT and Phoneme-level BERT can be directly used as an input encoder in a typical neural TTS system. Note that the success of these pre-trained language models has been limited to the English language only. Taking into account a societal, linguistic, cultural, machine learning and cognitive perspective [19], it is worth exploring pre-trained models for phoneme representations in languages other than English.

To fill the gap, we train the *first* large-scale multilingual language model for phoneme representations, using a pre-training corpus of 330M phonemic description sentences from nearly 100 languages and locales. Our model is trained based on the RoBERTa pre-training approach [11], using the BERT-base model configuration [10]. We conduct experiments on the downstream TTS task, directly employing our model as an input phoneme encoder of the strong model VITS [9]. Experimental results show that our model helps boost the performance of VITS, obtaining more natural prosody than the original VITS without pre-training and also producing fairly high-quality synthesized speech with limited training data. We summarize our contribution as follows:

- We present the first large-scale pre-trained multilingual model for phoneme representations, which we name XPhoneBERT.
- On the downstream TTS task, XPhoneBERT helps significantly improve the performance of the strong baseline VITS, thus confirming its effectiveness.
- We publicly release XPhoneBERT at `https://github.com/VinAIResearch/XPhoneBERT`. We hope that our XPhoneBERT model would help facilitate future research and downstream TTS applications for nearly 100 languages and locales.

10.21437/Interspeech.2023-444

2. Our XPhoneBERT

This section outlines the architecture and describes the multilingual pre-training corpus and optimization setup that we use for XPhoneBERT.

2.1. Model architecture

XPhoneBERT has the same model architecture as BERT-base [10]—a multi-layer bidirectional Transformer encoder [20]—in which the number of Transformer blocks, the hidden size and the number of self-attention heads are 12, 768 and 12, respectively. To pre-train XPhoneBERT, we use the masked language modeling objective [10] and follow the RoBERTa pre-training approach [11] which robustly optimizes BERT for better performance, i.e. using a dynamic masking strategy and without the next sentence prediction objective. Given the popularity of BERT and RoBERTa, we do not further detail about the architecture here. See [10, 11] for more information.

2.2. Multilingual pre-training data

Our multilingual pre-training dataset is constructed following three phases. The first phase is to collect text documents and then perform word and sentence segmentation as well as duplicate removal and text normalization. The second phase is to convert texts into phonemes, employing the CharsiuG2P toolkit [21] that supports 90+ languages and locales. Finally, the third phase is to perform phoneme segmentation.

2.2.1. First phase: Data collection and pre-processing

We collect texts for the languages supported by CharsiuG2P. Here, we employ the multilingual datasets `wiki40b` [22] and `wikipedia` [23], available to download from the Hugging Face *datasets* library [24]. In particular, we first download the `wiki40b` dataset consisting of text documents for 41 Wikipedia languages and locales.[1] We then use `wikipedia` to extract texts from Wikipedia dumps for remaining languages other than those belonging to `wiki40b`.[2]

We perform word and sentence segmentation on all text documents in each language by using the `spaCy` toolkit,[3] except for Vietnamese where we employ RDRSegmenter [25] from the VnCoreNLP toolkit [26]. We then lowercase all sentences and filter out duplicate sentences and single-word ones. We also apply text normalization to convert texts from their written form into their verbalized form for only English, German, Spanish, Vietnamese and Chinese (it is because we could not find an effective text normalization tool publicly available for other languages). Here, we use the text normalization component from the NVIDIA NeMo toolkit [27] for English, German, Spanish and Chinese, and the `Vinorm` text normalization package for Vietnamese.[4]

2.2.2. Second phase: Text-to-phoneme conversion

For each language whose locales do not have their own Wikipedia data,[5] we randomly divide the language's Wikipedia data into equal parts (each with the same number of sentences), with each part corresponding to a locale. For example, we divide 67 million English sentences into two equal parts that are then separately converted into phonemic descriptions in British English (`eng-uk`) and American English (`eng-us`).

Table 1: *Our pre-training data statistics. "LCode" denotes the ISO 639-3 code for each language or locale, while "#s" denotes the number of sentences.*

LCode	**#s (K)**	**LCode**	**#s (K)**	**LCode**	**#s (K)**
ady	2	glg	3793	ron	1816
afr	1793	grc	947	rus	15923
amh	73	gre	947	san	114
ara	2820	grn	60	slo	1143
arg	383	guj	211	slv	1167
arm-e	2989	hbs-cyrl	2007	sme	27
arm-w	175	hbs-latn	2007	snd	215
aze	3139	hin	287	spa	3936
bak	1272	hun	4372	spa-latin	3936
bel	2750	ice	776	spa-me	3936
ben	1785	ido	224	sqi	1373
bos	1464	ina	100	srp	2449
bul	1919	ind	2196	swa	537
bur	393	ita	12335	swe	5226
cat	4017	jam	8	tam	2289
cze	4542	jpn	12197	tat	984
dan	1714	kaz	1850	tgl	628
dut	7683	khm	93	tha	567
egy	3093	kor	2384	tts	567
eng-uk	33515	kur	335	tuk	105
eng-us	33515	lat-clas	597	tur	2148
epo	4333	lat-eccl	597	ukr	6967
est	1558	lit	1087	vie-c	2519
eus	3429	ltz	817	vie-n	2519
fas	1957	mac	2597	vie-s	2519
fin	4100	min	377	wel-nw	714
fra	11255	mlt	180	wel-sw	714
fra-qu	11255	ori	158	yue	908
geo	1211	pap	27	zho-s	6934
ger	33845	pol	7045	zho-t	6955
gla	121	por-bz	3437	_	_
gle	488	por-po	3437	_	_

To convert sentences into their phonemic description, we employ the grapheme-to-phoneme conversion toolkit CharsiuG2P [21]. The pre-trained CharsiuG2P is a strong multilingual Transformer-based model that generates the pronunciation of a word given its orthographic form and ISO 639-3 language code pair. Following the recommendation from [21], if the input word is in the CharsiuG2P toolkit's pronunciation dictionary of the target language/locale, we employ the pronunciation dictionary to generate the word's phonemic description. Otherwise, if the word is out of the vocabulary, we employ the pre-trained CharsiuG2P model to generate its phonemic description.

For example, given an input word "model" and the language code `eng-us` of American English, CharsiuG2P produces an output phoneme sequence of "ˈmɑdəɫ". Such an American English sentence as "a multilingual model" is thus converted into a phoneme sequence of "ˈeɪ ˌməɫtiˈɫɪŋwəɫ ˈmɑdəɫ". Note that in this conversion phase, we keep punctuations intact, as do the TTS systems [6, 7, 8, 9].

[1] `https://huggingface.co/datasets/wiki40b`

[2] `https://huggingface.co/datasets/wikipedia`

[3] `https://spacy.io`

[4] `https://github.com/v-nhandt21/Vinorm`

[5] Languages whose locales do not have their own Wikipedia data are: English (eng-uk & eng-us), French (fra & fra-qu), Greek (grc & gre), Latin (lat-clas & lat-eccl), Portuguese (por-po & por-bz), Serbo-Croatian (hbs-latn & hbs-cyrl), Spanish (spa & spa-latin & spa-me), Thai (tha & tts), Vietnamese (vie-n, vie-c & vie-s) and Welsh (wel-nw & wel-sw). By contrast, Armenian and Chinese have the corresponding Wikipedia data for their locales (Armenian: arm-e & arm-w; Chinese: min, yue, zho-s & zho-t).

```
from transformers import AutoModel, AutoTokenizer

tokenizer = AutoTokenizer.from_pretrained("vinai/xphonebert-base")
model = AutoModel.from_pretrained("vinai/xphonebert-base")
input_phonemes = "ˈe ɪ ▁ ˌm ə ɫ t i ˈɫ ɪ ŋ w ə ɫ ▁ ˈm  ɑ d ə ɫ"
input_ids = tokenizer(input_phonemes, return_tensors="pt")
features = model(**input_ids)
```

Figure 1: *An example code using XPhoneBERT for feature extraction with the Hugging Face `transformers` library in Python. Here, the input phonemes represent a phonemic description of the word-level sequence "a multilingual model".*

2.2.3. Third phase: Phoneme segmentation

CharsiuG2P converts each input word into a sequence of consecutive phonemes without a phoneme boundary indicator (e.g. white space). To better map between phonemes and speech [7, 9, 18], we would have to perform phoneme segmentation on the CharsiuG2P's output. Following [28], we employ the `segments` toolkit for phoneme segmentation.[6] Thus an input word is now converted into a sequence of phonemes separated by white spaces, e.g. "model" is converted into "ˈm ɑ d ə ɫ" in `eng-us`. Since we use the white space to separate phonemes, to distinguish phonemes belonging to different word tokens, we employ a meta symbol ▁ (U+2581) for marking word boundaries. For example, the American English sentence "a multilingual model" is now converted into the phoneme-segmented sequence "ˈe ɪ ▁ ˌm ə ɫ t i ˈɫ ɪ ŋ w ə ɫ ▁ ˈm ɑ d ə ɫ".[7]

2.2.4. Pre-training data statistics

Through the 3-phase construction process, we finally obtain a pre-training corpus of 330M phoneme-level sentences across 94 languages and locales. We present the data statistic for each language or locale in Table 1.

2.3. Optimization

We employ a white-space tokenizer, resulting in a vocabulary of 1960 phoneme types. Our XPhoneBERT thus has a total of 87.6M parameters. For training XPhoneBERT on our multilingual pre-training corpus, we employ the RoBERTa implementation [11] from the `fairseq` library [29]. We set a maximum sequence length of 512. We optimize the model using Adam [30] and use a batch size of 1024 sequence blocks across 8 A100 GPUs (40GB each) and a peak learning rate of 0.0001. We train for 20 epochs in about 18 days (here, the first 2 epochs are used for warming up the learning rate).

2.4. Usage example

To show the potential use for downstream tasks, we present in Figure 1 a basic usage of our pre-trained model XPhoneBERT for feature extraction with the `transformers` library [31]. More usage examples of XPhoneBERT can be found at the XPhoneBERT's GitHub repository.

3. Experimental setup

We evaluate the effectiveness of XPhoneBERT on the downstream text-to-speech (TTS) task. Due to a limited resource of human raters, we perform this TTS task for American English (`eng-us`) and Northern Vietnamese (`vie-n`).[8]

3.1. TTS datasets

For English, we use the benchmark dataset LJSpeech [32] consisting of 13,100 audio clips of a single speaker with a total duration of about 24 hours (here, each clip is also provided with a gold-standard text transcription). Following [9], the dataset is split into training, validation and test sets of 12,500, 100 and 500 clip samples, respectively.

For Vietnamese, we randomly sample 12,300 different medium-length sentences from the PhoBERT pre-training news data [33]. We hire a professional speaker to read each sentence in a studio and record the corresponding audio, resulting in a total duration of about 18 hours for 12,300 high-quality audio clips. We split our Vietnamese TTS dataset into training, validation and test sets of 12,000, 100 and 200 clips, respectively.

3.2. TTS modeling and training

We employ the strong TTS model VITS [9].[9] VITS is an end-to-end model that contains a Transformer encoder [20] to encode the input phoneme sequence. We extend VITS with XPhoneBERT by replacing the VITS's Transformer encoder with XPhoneBERT.

For the first setting of using the whole TTS training set, we train the original VITS model with optimal hyper-parameters used in its paper [9], e.g. using the AdamW optimizer [34] with $\beta_1 = 0.8, \beta_2 = 0.99$ and the weight decay $\lambda = 0.01$, and an initial learning rate of 2×10^{-4} (here, the learning rate decay is scheduled by a $0.999^{1/8}$ factor in every epoch). We run for 300K training steps with a batch size of 64 (i.e. equivalent to about 1600 training epochs for both English and Vietnamese). For training the VITS variant extended with XPhoneBERT, we apply the same training protocol used for the original VITS. Here, XPhoneBERT is frozen in the first 25% of the training steps and then updated during the remaining training steps.

We also experiment with another setting where the TTS training data is limited. In particular, for each language, we

[6] `https://pypi.org/project/segments`

[7] For convenience, we also create a Python package named `text2phonemesequence`, incorporating both CharsiuG2P and `segments`, to perform a direct conversion from an input word-level sentence (e.g. "a multilingual model") to an output phoneme-segmented sequence (e.g. "ˈe ɪ ▁ ˌm ə ɫ t i ˈɫ ɪ ŋ w ə ɫ ▁ ˈm ɑ d ə ɫ").

[8] The model weights of PnG BERT (`https://google.github.io/tacotron/publications/png_bert`), Mixed-Phoneme BERT (`https://speechresearch.github.io/mpbert`) and Phoneme-level BERT (`https://github.com/yl4579/PL-BERT`) are not published at the time of our empirical investigation (here, these pre-trained models are still not yet publicly available on 8th March 2023—the INTERSPEECH 2023's paper update deadline). Therefore, we could not compare our multilingual XPhoneBERT with those monolingual models for English.

[9] `https://github.com/jaywalnut310/vits`

Table 2: *Obtained results on the English test set. "100%" and "5%" denote the first experimental setting of using the whole TTS training set and the second experimental setting of using only 5% of the TTS training set for training, respectively. "XPB" abbreviates our XPhoneBERT. The MOS is reported with 95% confidence intervals (here, each MOS score difference between two models is significant with p-value < 0.05).*

	Model	**MOS (↑)**	**MCD (↓)**	**RMSE$_{\mathbf{F_0}}$ (↓)**
	Ground truth	4.39 ± 0.08	0.00	0.00
100%	Baseline VITS	4.00 ± 0.08	7.04	377
	VITS w/ XPB	**4.14** ± 0.07	**6.63**	**348**
5%	Baseline VITS	2.88 ± 0.11	7.40	407
	VITS w/ XPB	**3.22** ± 0.11	**7.15**	**383**

Table 3: *Obtained results on the Vietnamese test set (here, each MOS score difference between two models is significant with p-value < 0.05).*

	Model	**MOS (↑)**	**MCD (↓)**	**RMSE$_{\mathbf{F_0}}$ (↓)**
	Ground truth	4.26 ± 0.06	0.00	0.00
100%	Baseline VITS	3.74 ± 0.08	5.41	249
	VITS w/ XPB	**3.89** ± 0.08	**5.12**	**234**
5%	Baseline VITS	1.59 ± 0.05	6.20	291
	VITS w/ XPB	**3.35** ± 0.10	**5.39**	**248**

randomly sample 5% of the training audio clips, and then only use those sampled audios for training (total duration of about 1.2 hours for English and about 0.9 hours for Vietnamese). We apply the same training protocol used for the first setting with an exception that we run for 100K training steps.

3.3. Evaluation protocol

We evaluate the performance of TTS models using subjective and objective metrics. For subjective evaluation of the naturalness, for each language, following [2, 7, 9], we randomly select 50 ground truth test audios and their text transcription to measure the Mean Opinion Score (MOS). Here, for each text transcription, we synthesize speeches using 4 different models (including the baseline VITS and the VITS variant extended with our XPhoneBERT, which are trained under the two different experimental settings detailed in the previous subsection). For each language, we hire 10 native speakers to rate each of the five speeches (i.e. the four synthesized speeches and the ground truth speech) on a naturalness scale from 1 to 5 with 1-point increments. Here, each rater does not know which model produces which speech.

For objective evaluations of the distortion and intonation difference between the ground truth speech and the synthesized speech, we compute two metrics of the mel-cesptrum distance (MCD; dB) and the F0 root mean square error (RMSE$_{\mathrm{F_0}}$; cent), according to the implementation from [35].

4. Main results

Tables 2 and 3 show obtained results for English and Vietnamese, respectively. We find that our XPhoneBERT helps improve the performance of VITS on all three evaluation metrics for both English and Vietnamese in both experimental settings. For example, for the first setting of using the whole TTS training set for training, the MOS score significantly increases from 4.00 to 4.14 (+0.14 absolute improvement) for English and from 3.74 to 3.89 (+0.15) for Vietnamese. When it comes to the second setting of using limited training data, XPhoneBERT helps produce larger absolute MOS improvements than those for the first setting. That is, MOS increases from 2.88 to 3.22 (+0.34) for English and especially from 1.59 to 3.35 (+1.76) for Vietnamese, clearly showing the effectiveness of XPhoneBERT.

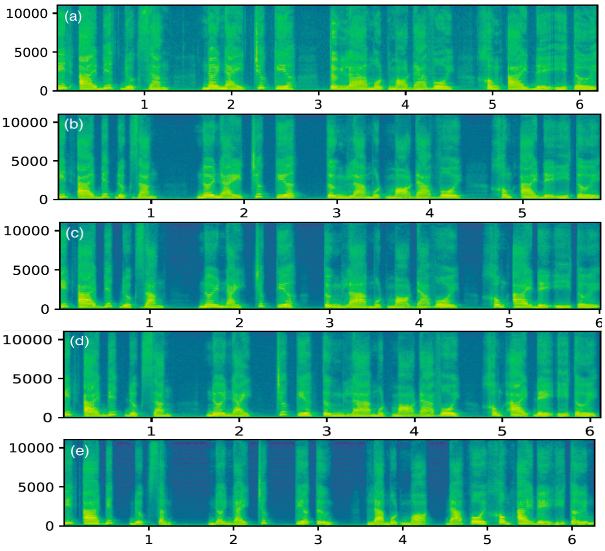

Figure 2: *Spectrograms visualization by different models. The text of the speech is "Ít ai biết được rằng nơi này trước kia từng là một mỏ đá vôi không ai để ý tới" (Little is known that this place was once a limestone quarry that no one paid any attention to). (a): Ground truth; (b): VITS with XPhoneBERT, under the first experimental setting; (c): VITS with XPhoneBERT, under the second experimental setting; (d): Original VITS, under the first setting; (e): Original VITS, under the second setting.*

Similar to [36], we also find that the subjective evaluation metric MOS is not "always" correlated with the objective evaluation metrics MCD and RMSE$_{\mathrm{F_0}}$. That is, for Vietnamese in Table 3, the baseline VITS under the first setting obtains higher MOS but slightly poorer MCD and RMSE$_{\mathrm{F_0}}$ than the VITS extended with XPhoneBERT under the second setting (MOS: 3.74 vs. 3.35; MCD: 5.41 vs. 5.39; RMSE$_{\mathrm{F_0}}$: 249 vs. 248).

From obtained results for the baseline VITS under the first setting and the VITS extended with XPhoneBERT under the second setting in both Tables 2 and 3, we might consider that XPhoneBERT helps synthesize fairly high-quality speech with limited training data. We also visualize the spectrograms of synthesized and ground truth speeches for a Vietnamese text transcription in Figure 2, illustrating that XPhoneBERT helps improve the spectral details of the baseline VITS's output.

5. Conclusion

We have presented the first large-scale multilingual language model XPhoneBERT pre-trained for phoneme representations. We demonstrate the usefulness of XPhoneBERT by showing that using XPhoneBERT as an input phoneme encoder improves the quality of the speech synthesized by a strong neural TTS baseline. XPhoneBERT also helps produce fairly high-quality speech when the training data is limited. We publicly release XPhoneBERT and hope that it can foster future speech synthesis research and applications for nearly 100 languages and locales.

6. References

[1] J. Kong, J. Kim, and J. Bae, "HiFi-GAN: Generative Adversarial Networks for Efficient and High Fidelity Speech Synthesis," in *Proceedings of NeurIPS*, 2020.

[2] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "FastSpeech: Fast, Robust and Controllable Text to Speech," in *Proceedings of NeurIPS*, 2019.

[3] V. Popov, I. Vovk, V. Gogoryan, T. Sadekova, and M. Kudinov, "Grad-TTS: A Diffusion Probabilistic Model for Text-to-Speech," in *Proceedings of ICML*, 2021, pp. 8599–8608.

[4] X. Tan, J. Chen *et al.*, "NaturalSpeech: End-to-End Text to Speech Synthesis with Human-Level Quality," *arXiv preprint*, vol. arXiv:2205.04421, 2022.

[5] J. Shen, R. Pang *et al.*, "Natural TTS Synthesis by Conditioning Wavenet on MEL Spectrogram Predictions," in *Proceedings of ICASSP*, 2018, pp. 4779–4783.

[6] N. Li, S. Liu, Y. Liu, S. Zhao, and M. Liu, "Neural Speech Synthesis with Transformer Network," in *Proceedings of AAAI*, 2019, pp. 6706–6713.

[7] J. Kim, S. Kim, J. Kong, and S. Yoon, "Glow-TTS: A Generative Flow for Text-to-Speech via Monotonic Alignment Search," in *Proceedings of NeurIPS*, 2020.

[8] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "FastSpeech 2: Fast and High-Quality End-to-End Text to Speech," in *Proceedings of ICLR*, 2021.

[9] J. Kim, J. Kong, and J. Son, "Conditional Variational Autoencoder with Adversarial Learning for End-to-End Text-to-Speech," in *Proceedings of ICML*, 2021, pp. 5530–5540.

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of NAACL*, 2019, pp. 4171–4186.

[11] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," *arXiv preprint*, vol. arXiv:1907.11692, 2019.

[12] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A lite BERT for self-supervised learning of language representations," in *Proceedings of ICLR*, 2020.

[13] T. Hayashi, S. Watanabe, T. Toda, K. Takeda, S. Toshniwal, and K. Livescu, "Pre-Trained Text Embeddings for Enhanced Text-to-Speech Synthesis," in *Proceedings of INTERSPEECH*, 2019, pp. 4430–4434.

[14] T. Kenter, M. Sharma, and R. Clark, "Improving the Prosody of RNN-based English Text-To-Speech Synthesis by Incorporating a BERT model," in *Proceedings of INTERSPEECH*, 2020, pp. 4412–4416.

[15] G. Xu, W. Song, Z. Zhang, C. Zhang, X. He, and B. Zhou, "Improving Prosody Modelling with Cross-Utterance BERT Embeddings for End-to-end Speech Synthesis," in *Proceedings of ICASSP*, 2021, pp. 6079–6083.

[16] Y. Jia, H. Zen, J. Shen, Y. Zhang, and Y. Wu, "PnG BERT: Augmented BERT on Phonemes and Graphemes for Neural TTS," in *Proceedings of INTERSPEECH*, 2021, pp. 151–155.

[17] G. Zhang, K. Song, X. Tan, D. Tan, Y. Yan, Y. Liu, G. Wang, W. Zhou, T. Qin, T. Lee, and S. Zhao, "Mixed-Phoneme BERT: Improving BERT with Mixed Phoneme and Sup-Phoneme Representations for Text to Speech," in *Proceedings of INTERSPEECH*, 2022, pp. 456–460.

[18] Y. A. Li, C. Han, X. Jiang, and N. Mesgarani, "Phoneme-Level BERT for Enhanced Prosody of Text-to-Speech with Grapheme Predictions," *arXiv preprint*, vol. arXiv:2301.08810, 2023.

[19] S. Ruder, "Why You Should Do NLP Beyond English," https://ruder.io/nlp-beyond-english/, 2020. [Online]. Available: https://ruder.io/nlp-beyond-english/

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is All you Need," in *Proceedings of NIPS*, 2017, pp. 5998–6008.

[21] J. Zhu, C. Zhang, and D. Jurgens, "ByT5 model for massively multilingual grapheme-to-phoneme conversion," in *Proceedings of INTERSPEECH*, 2022, pp. 446–450.

[22] M. Guo, Z. Dai, D. Vrandečić, and R. Al-Rfou, "Wiki-40B: Multilingual Language Model Dataset," in *Proceedings of LREC*, 2020, pp. 2440–2452.

[23] W. Foundation. Wikimedia downloads. [Online]. Available: https://dumps.wikimedia.org

[24] Q. Lhoest, A. Villanova del Moral, Y. Jernite *et al.*, "Datasets: A Community Library for Natural Language Processing," in *Proceedings of EMNLP: System Demonstrations*, 2021, pp. 175–184.

[25] D. Q. Nguyen, D. Q. Nguyen, T. Vu, M. Dras, and M. Johnson, "A Fast and Accurate Vietnamese Word Segmenter," in *Proceedings of LREC*, 2018, pp. 2582–2587.

[26] T. Vu, D. Q. Nguyen, D. Q. Nguyen, M. Dras, and M. Johnson, "VnCoreNLP: A Vietnamese Natural Language Processing Toolkit," in *Proceedings of NAACL: Demonstrations*, 2018, pp. 56–60.

[27] O. Kuchaiev, J. Li *et al.*, "NeMo: a toolkit for building AI applications using Neural Modules," in *Proceedings of NeurIPS Workshop on Systems for ML*, 2019.

[28] M. Bernard and H. Titeux, "Phonemizer: Text to Phones Transcription for Multiple Languages in Python," *Journal of Open Source Software*, vol. 6, no. 68, p. 3958, 2021.

[29] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli, "fairseq: A Fast, Extensible Toolkit for Sequence Modeling," in *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019, pp. 48–53.

[30] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Proceedings of ICLR*, 2015.

[31] T. Wolf, L. Debut *et al.*, "Transformers: State-of-the-Art Natural Language Processing," in *Proceedings of EMNLP 2020: System Demonstrations*, 2020, pp. 38–45.

[32] K. Ito and L. Johnson, "The LJ Speech Dataset," https://keithito.com/LJ-Speech-Dataset/, 2017.

[33] D. Q. Nguyen and A. T. Nguyen, "PhoBERT: Pre-trained language models for Vietnamese," in *Findings of EMNLP*, 2020, pp. 1037–1042.

[34] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," in *Proceedings of ICLR*, 2019.

[35] A. Tamamori, T. Hayashi, K. Kobayashi, K. Takeda, and T. Toda, "Speaker-dependent WaveNet vocoder," in *Proceedings of INTERSPEECH*, 2017, pp. 1118–1122.

[36] T. Saeki, K. Tachibana, and R. Yamamoto, "DRSpeech: Degradation-Robust Text-to-Speech Synthesis with Frame-Level and Utterance-Level Acoustic Representation Learning," in *Proceedings of INTERSPEECH*, 2022, pp. 793–797.