Fast Bilingual Grapheme-To-Phoneme Conversion

Hwa-Yeon Kim , Jong-Hwan Kim , Jae-Min Kim
NAVER Corp., South Korea
{hwayeon.kim, jhwan.kim, kjm.kim} @navercorp.com

Abstract

Autoregressive transformer (ART)-based grapheme-to-phoneme (G2P) models have been proposed for bi/multilingual text-to-speech systems. Although they have achieved great success, they suffer from high inference latency in real-time industrial applications, especially processing long sentence. In this paper, we propose a fast and high-performance bilingual G2P model. For fast and exact decoding, we used a non-autoregressive structured transformer-based architecture and data augmentation for predicting output length. Our model achieved better performance than that of the previous autoregressive model and about 2700% faster inference speed.

1 Introduction

Speech synthesis has been applied in various real-world services, such as AI speaker, car navigation guidance and news article-reading services in each language. Grapheme-to-phoneme (G2P) module convert text to phonemes in text-to-speech (TTS) system. G2P conversion has been studied in various ways, including rules, dictionaries, statistical-based methods (Deri and Knight, 2016) and neural network-based methods (Yolchuyeva et al., 2021; Sun et al., 2019a; Kim et al., 2021; Choi et al., 2021). Currently, monolingual G2P research is the most conducted, although recently bilingual or multilingual G2P research is also being actively performed (Clematide and Makarov, 2021; Yu et al., 2020; Bansal et al., 2020; Gautam et al., 2021). Most of the proposed models with high performance are based on autoregressive transformers (A.Vaswani et al., 2017) in both monolingual and multilingual G2P. However, these models suffer from high inference latency, which is sometimes unacceptable for real-time TTS applications that generate long speech synthesis sounds, such as news sentences. A previous study (Kim et al., 2021) used a simple model structure with a few features and batch inference for fast inference speed; however, there were limitations in specific language characteristics.

In this paper, we propose a high-performance bilingual G2P model that has an fast inference speed that enables real-time service. For an efficient expression for each language, byte-level representation input and a language index are used as the main inputs, and for fast decoding, the transformer model is based on a non-autoregressive structured decoder. Because the length of the estimated output used in the non-autoregressive structured decoder has a great impact on the G2P accuracy, a sub-network and a data augmentation technique are used to better infer the output length. In addition, we experimented with the difference between training the whole input unit (sentence) and the tokenized unit.

We conducted experiments for different language systems, such as European which have a small number of graphemes and East Asian ones which have a large number of graphemes. We chose two languages for bilingual G2P model; English and Korean. Experimental results showed that, despite significantly losing speed, our non-autoregressive transformer-conditional random field (NART-CRF) based G2P model achieved better performance than those of previous ART models. When it is applied to an actual service system, in addition to the speed and high accuracy applicable to real-world TTS applications, it is possible to generate the phonemes of several languages with one model.

2 Related work

2.1 Multilingual G2P

Recent works propose various methods for multilingual natural language processing (NLP) tasks such as machine translation (Aharoni et al., 2019; Zhang et al., 2020) and language model (Pires et al., 2019). A few multilingual G2P studies are also in

Proceedings of NAACL-HLT 2022: Industry Track Papers, pages 289 - 296

progress. The benchmarks for multilingual g2p is provides and utilized various G2P models : A neural transducer system using an imitation learning paradigm (Ashby et al., 2021), studies building an ensemble of several different sequence models (Vesik et al., 2020; Gautam et al., 2021; Clematide and Makarov, 2021). Meanwhile, there is a neural multilingual G2P model with byte-level input representation (Yu et al., 2020). On this wise, most of the autoregressive sequence models are used to learn phonemes of various languages. But, the autoregressive factorization makes the inference process hard to be parallelized as the results are generated token by token sequentially. Therefore, these models have limitations in applying them to real-world processing services, especially dealing with long sentence, because the inference time increases linearly with the length of the generated phoneme output.

2.2 Fast decoding

For various tasks, the transformer (A.Vaswani et al., 2017) model achieve good performance. However, the autoregressive method suffer from high inference latency. Therefore, there are several studies to solve this problem. Since decoding takes a high inference latency, the deep-encoder and shallow-decoder architecture is proposed and it improve the inference speed (Kasai et al., 2021). For parallelism, the non-autoregressive sequence models are proposed and applied it to the machine translation and speech synthesis (Gu et al., 2018; Sun et al., 2019b). The non-autoregressive sequence models improve the inference speed; however, they cannot get results as good as their autoregressive counterparts that generate each token in the target sentence independently. To decode token co-occurrence be guaranteed, a structured inference module is incorporated in the non-autoregressive decoder to directly model the multi-modal distribution of phoneme sequences (Sun et al., 2019b). In this study, we follow the structure (Sun et al., 2019b) to apply G2P task and achieve great performance.

3 The proposed model

This section describes the proposed model for fast bilingual G2P conversion. The overall structure of the model is shown in Figure 1.

3.1 Byte-level representation input and sentence/token-level input

Following the method of Yu et al. (2020), the proposed model uses an input with a byte-level representation for the efficient representation of multiple languages. Each character is expressed at the byte level based on the UTF-8 encoding. This expression can reduce the size of the input vocabulary, and the byte-level vocabulary cardinality is constrained to be equal to or smaller than 256. In this study, two experiments were performed: processing of the entire sentence as the input, and tokenizing of the sentence and processing of each token as one batch.

Processing of the entire sentence as the input : The input sentence encoded at the byte level and the language index of the input are used as the inputs to the model. Using the entire sentence as the input is good for inferring the correct pronunciation sequence according to the meaning because it learns by considering the context of the entire sentence together. On the other hand, if the dataset is divided by language, as in this experiment, it is necessary to separate and process the language-mixed sentences for each language when inferring the pronunciation sequence.

Processing of the input token unit : First, a given input sentence is divided into tokens using an appropriate tokenizer for the language. In the case of Korean and English, a tokenizer that separates the space-delimited orthographic words (tokens) was used in this study. Here, in the case of Korean, there is a point to be particularly careful about. The pronunciation of the first syllable or the last syllable of a token may change depending on whether the tokens are read after a break or not. Therefore additional features were needed to connect the separated tokens naturally in the final G2P results. Following the method of Kim et al. (2021), we used the phonological phrasing information between tokens. Moreover, for the first and last syllables of token to be naturally connected with each front/next token, information on the ending or beginning of the part to be connected is required. For example, for the input as shown in the Figure 2, each token's input elements in the input sentence are as follows: A language index, a input token x_t to be converted to a byte-level representation (part *a*), two phonological phrasing information on both sides of the token (part *b*), a last *jaso* (orthographic phoneme segments) in front token x_{t-1} (part *c*) and a first

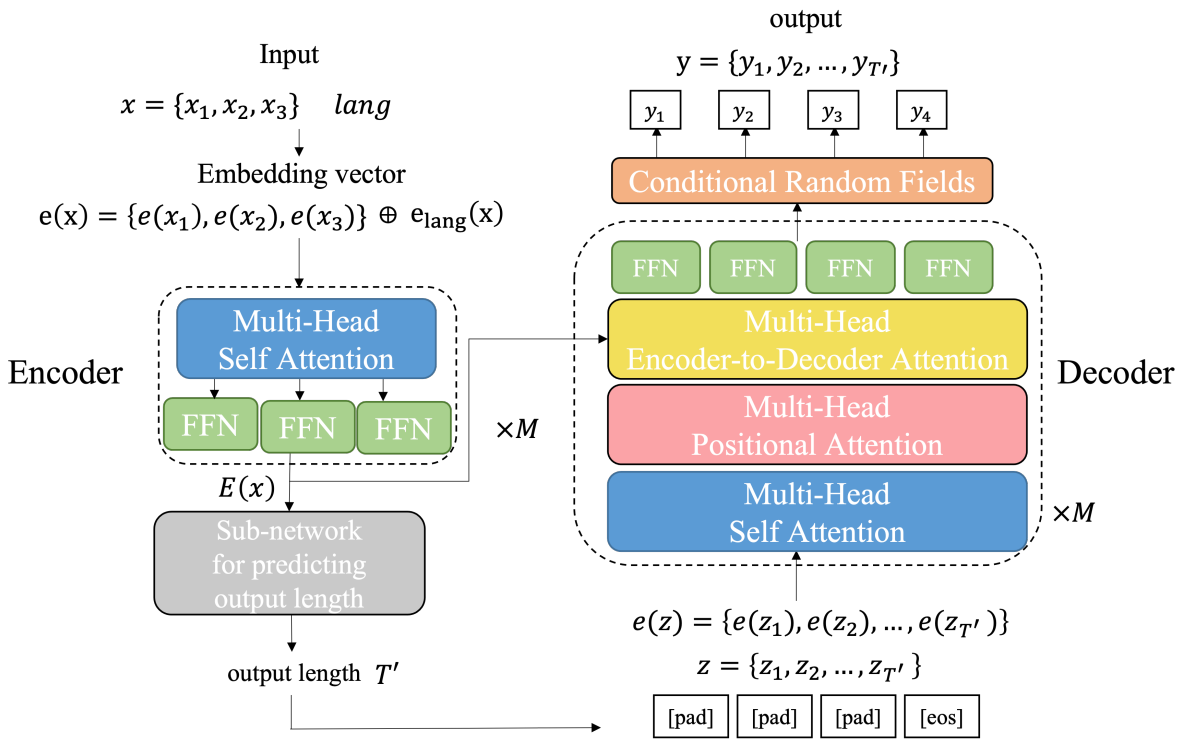

Figure 1: The overview of proposed model

~ 모바일^{c} < *clitic* >^{b} 앱을^{a} < *ap* >^{b} 통^{d} 해서 ~

(Means : through the mobile app)

Language index = Korean

Figure 2: The example of composition for each token in a sentence

jaso in next token x_{t+1} (part d). They are concatenated with each token, and the entire token in the sentence is composed of one batch, so that it learns and infers at once. In this way, if it is configured in token units, it is not necessary to separate the language-mixed sentences for each language and compose the input, and it is possible to infer faster with relatively short input and output lengths. On the other hand, tokenizers for each language are required, and there is a limit for including context information rather than the entire sentence unit.

3.2 Transformer-based structured decoding model for G2P conversion

The model design follows the NART architecture with CRFs. For more information on the model, see A.Vaswani et al. (2017); Gu et al. (2018); Sun et al. (2019b).

NART-based model : Like in the ART model, the encoder of our model takes the embeddings of the input tokens and their additional features as the input and generates a contextual representation. Following the decoder in NART-CRF, the decoder independently decodes each pronunciation token given a sequence length T' and a decoder input z. It also uses the padding symbol "<pad>" followed by the end-of-sentence symbol "<eos>" as the decoder input. The transformer model utilizes multi-head self-attention and multi-head encoder-decoder attention. In contrast to the ART model, multi-head positional attention in the decoder is also used to model local word orders within a sentence or a token. In our model, each decoder layer refers to the output of each encoder layer with the same depth. It follows the model architecture of Yu et al. (2020) and performs better than the existing architecture in our experiment. The position-wise feedforward network consists of a two-layer linear transformation with a ReLU activation function and is applied after using multi-head attention in both the encoder and the decoder.

Structured inference module: Like in Sun et al. (2019b), a linear-chain CRF is incorporated into the decoder part to model richer structural dependencies. The CRF module can be jointly trained end to end with neural networks using a negative log-likelihood loss L_{CRF}. In the context of G2P

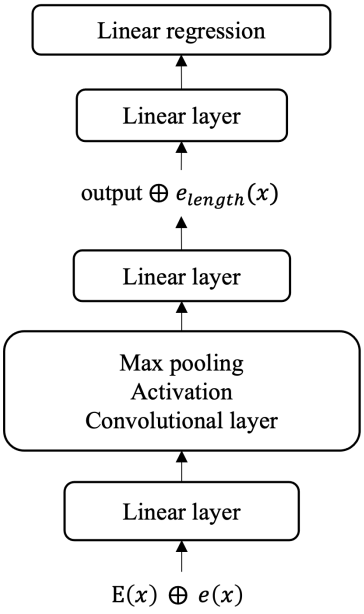

Figure 3: The sub-network for predicting and using output length

conversion, we use a "phoneme" for the decoder output and decode its highest scoring sequence.

3.3 Predicting output length for decoder

In the NART-CRF structure, an input of a specific length is used as the input z of the decoder. The length of this input has a great influence on inferring the final output of the model. Through several experiments, we realized that it is not easy to predict the exact output length using only the encoder output. Even if it is long or short by a small number such as 1 or 2, the pronunciation sequence can be generated incorrectly, which greatly affects the performance. So, while adding a layer or sub-network to predict the output length T', we applied a data augmentation technique that can supplement the decoding process despite incorrect prediction values.

Sub-network for predicting the output length In the G2P task, the prediction of the input and output lengths of the decoder has a greater effect on the overall accuracy than that in the machine translation task (Sun et al., 2019b). We added a sub-network to infer the phoneme sequence length exactly, as shown in Figure 3. The sub-network follows the model proposed in Yang et al. (2020); however, it differs in the prediction of an output length that is continuous in nature using linear regression rather than softmax at the end of the model.

Data augmentation As mentioned above, the length of the sentence is very important in the phoneme sequence of the G2P model. Therefore, even if the sentence length is incorrectly predicted, it should still be used to generate a phoneme sequence with the correct length. Thus, we trained model to guess correctly actual output length by padding by the length that exceeds the actual length even in a sequence that is a little longer than the actual output length. To this end, data augmentation was performed by pairing an output with an output length of 1 or 2 longer in addition to the existing dataset and filled with a padding tag with an existing input.

Joint training with regression loss: Our training loss L is the sum of the CRF negative log-likelihood loss L_{CRF} and the mean square error (MSE) of the sub-network as loss L_{length}. :

$$L = L_{CRF} + L_{length} = -\log P(y|x) + (T - T')^2 \quad (1)$$

4 Experiments

4.1 Experimental settings

We collected scripts of domains used in real-world services and constructed a Korean and English G2P dataset by labeling it from speech. A voice actor read a Korean or English script naturally, and taggers dictated the phonological phrasing information and pronunciations exactly as they heard them. We used 20,000 sentences in each language for training and 200 samples in each language for testing. Each sentence consisted of an average of 12.45 tokens (words in English and *Eojoel* in Korean) and the average length of output for each token is 5 and the maximum is 29. The phonological phrasing information used in this model is mainly composed of the intonation phrase (IP), accent phrase (AP), clitic, and end of sentence (sb). IP refers to reading with a pause, and AP refers to a delimitation. The size of input vocabulary of bilingual was 110 and the number of phonemes was 42 in Korean and 39 in English. We used the default network architecture of the original base transformer (A.Vaswani et al., 2017), which consists of a four-layer encoder and a four-layer decoder.

4.2 Inference

In the training process, to generate an accurate phoneme sequence, we performed data augmentation so that the pad was filled even when a length

exceeding the actual length was predicted. In fact, the model predicted a length that was a few smaller or longer than the actual output length. So, we bias the predicted length so that the decoder's input is made longer than the actual output length in most cases. It is intended that the pad will eventually be filled in to generate a phoneme sequence of the correct length.

We evaluate the average per-sentence decoding latency with a single NVIDIA Tesla V100 GPU for the ART-G2P and our models to measure the speedup.

4.3 Evaluation

The evaluation metrics used in the experiment were the phoneme error rate (PER), accuracy (Acc) and accuracy of length (L-Acc). PER, as used in the evaluation of the G2P model performance (Yu et al., 2020) , is the Levenshtein distance between the predicted phoneme sequences and the reference phoneme sequences, divided by the number of phonemes in the reference pronunciation. Acc is the percentage of sentences in which the predicted phoneme sequence exactly matches the reference pronunciation. L-Acc is the percentage of length in which the predicted phoneme sequence exactly matches the reference pronunciation sequence's length.

4.4 Results : ART vs NART

Table 1 shows the performance of the ART (Yu et al., 2020) and the proposed G2P model with a sentence- or token-level input. While ART-G2P shows high accuracy, the inference time is very long. When time was measured for each area, the average encoding and decoding time was 40/66ms, but since ART continuously decodes as much as the output length, the time increases linearly as much as the output length. On the other hand, the proposed NART-CRF based model trained at sentence-level showed about 22 times faster speed than ART-G2P; but, it was less accurate than ART-G2P. The model trained in token unit showed higher accuracy with about 27 times faster inference speed, confirming that it is a fast and accurate model structure.

It is analyzed that the proposed model has outperforms ART in the Korean dataset, because it refers to the phonological phrasing information. In the case of the proposed model, the token-level showed higher performance in both languages because the shorter input length is more advantageous in predicting the output length. When looking at the distribution of the difference between the actual length and the predicted length, in the case of sentence units, there was a large deviation, which caused a lot of errors.

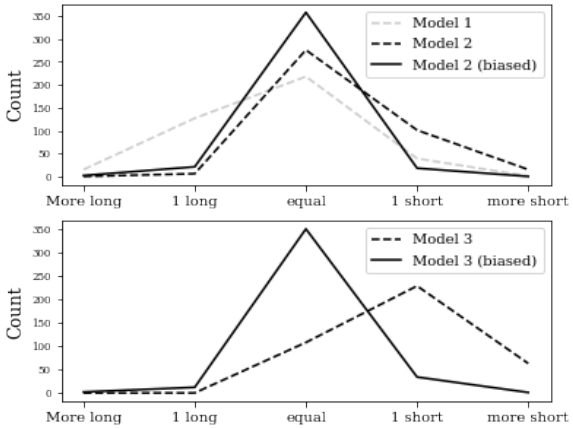

Figure 4: Results with predicted output length (biased or not)

4.5 Ablation study about augmentation

The Table 2 is an ablation study showing whether the method described in Section 3.3 is effective. The compared models are three models trained at sentence-level : *Model 1* incorporating regression layer for predicting output length in NART-CRF, *Model 2* trained with data augmentation in the same structure as Model1, *Model 3* incorporating sub-network for predicting output length and trained with data augmentation. Figure 4 shows how much the predicted sentence length differs from the actual sentence length. Looking at the sentence length prediction result of Model 1, it is inferred a lot with approximations around the actual sentence length, so the sentence length accuracy is only 54.5%. Model 2 has a slightly higher value for accurately predicting the length than Model 1. Through this, it can be seen that data augmentation is effective in accurately predicting the length of a sentence by filling the "<pad>" tag even in sentences that are longer than the actual length. However, since data augmentation was performed only in cases of be longer, there are still cases in which it is not applied for shorter than actual length. Therefore we used the predicted sentence length with a bias of 2, and actually showed a big increase in performance. In the case of Model 3, the accuracy of length was very low at 27% because the sentence length was often predicted shorter than the actual length, but when the sentence length was

Model	Language	Acc (%)	PER (%)	Inference time (ms/sent)
ART-G2P	Merged	83.25	0.62	3830
	English	92.50	0.43	
	Korean	74.00	0.82	
Sentence-level NART-CRF G2P	Merged	81.00	0.64	177.15 ($\times$22)
	English	84.50	0.72	
	Korean	77.50	0.56	
Token-level NART-CRF G2P	Merged	**87.75**	**0.43**	**140** ($\times$27)
	English	**93.00**	**0.38**	
	Korean	**82.50**	**0.49**	

Table 1: The table shows results of the ART-G2P and proposed NART-CRF G2P models with sentence and token level training. We evaluate accuracy, PER of model and inference time in each language.

Model	Acc (%)	PER (%)	L-Acc (%)
Model 1 ; NART-CRF	48.75	2.91	54.5
Model 2 ; NART-CRF + augm	63.75	1.48	69.0
Model 2 + biased	**81.50**	0.80	**89.5**
Model 3 ; NART-CRF w/subNN + augm	24.50	4.22	27.0
Model 3 + biased	81.00	**0.64**	**87.8**

Table 2: The Ablation study about data augmentation and bias

biased during inference, the length prediction accuracy increased significantly. In fact, looking at the generated result, when the actual sentence length is 14 and the biased inference sentence length is 17, the pronunciation sequence is generated as $y = \{y_0, y_1, ..., y_{13}, pad, pad, pad\}$. If "<pad>" tags are deleted in post-processing, the inference result and the correct answer were matched. The proposed method of biasing the sentence length predicted in inference and data augmentation make predict the correct length through an additional decoding process even at the predicted length as an approximation of the actual sentence length. The proposed method of biasing the sentence length predicted in inference and data augmentation make predict the correct length through an additional decoding process even at the predicted length as an approximation of the actual sentence length.

4.6 In real-time TTS application

We applied it to the industrial TTS system. In our system, bilingual TTS attempts to generate a pronunciation sequence based on a specific language for an input with mixed languages. To this end, numbers and symbols are normalized based on a specific language, and each language goes through processing such as estimation of phonological phrasing information for each language. In bilingual G2P, the phoneme sequence is generated with the grapheme processed for each language for the input with mixed languages and then connect the results.

We utilized the Open Neural Network Exchange (ONNX) [1] to apply to a TTS system running in a CPU environment[2]. ONNX is an open-source machine-independent format and widely used for exchanging neural network models. First, our model implemented in tensorflow was exported to ONNX format, and inference was performed using Onnxruntime [3]. Onnxruntime is a cross-platform inference and training machine-learning accelerator. It performs hardware acceleration through graph optimization, graph partition and then distributed runner.

We applied our model to a real-time processing system and inferred at an average speed of $40ms/sent$ for 1000 sentences. In addition, we measured the Real Time Factor (RTF) when only the monolingual G2P module used in the existing system was changed to our model. As our Unit-selection Text-to-Speech (UTS) system, it is judged that real-time processing is possible only when the volume of processing is less than 0.1RT. When 500 sentences

[1] https://github.com/onnx/onnx

[2] Intel(R) Xeon(R) Silver 4114 CPU @ 2.20GHz (40 cores)

[3] https://onnxruntime.ai

were processed for each language, 0.026 to 0.037 RTx for Korean and 0.033 to 0.057 RTx for English were measured, confirming that real-time processing was possible.

5 Conclusion

In this study, a structure of a NART-CRF was proposed for fast bilingual G2P with real-time processing. For bilingual, input of byte representation was used, and additional sub-network and data augmentation techniques were used for accurate output length inference. The proposed model showed higher accuracy than the existing ART-G2P and at the same time showed about 27 times faster inference speed. In addition, when applied to an industrial TTS system, the speed was improved to a level capable of real-time processing.

In future work, we will study a model with contextual information or representation of language model to solve some error cases caused by lack of context. Furthermore, we will experiments with fast "multilingual" G2P by expanding the language types to Chinese, Japanese, and European languages. As a result of testing two different language system (i.e. European and East Asian), It is expected that expansion of languages, which others in same language group, will be possible. Additionally, considering the accents and tones used in languages such as English and Chinese, and training on an unbalanced dataset remain issues to be resolved.

References

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *proc. Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

Lucas F.E. Ashby, Travis M. Bartley, Simon Clematide, Luca Del Signore, Cameron Gibson, Kyle Gorman, Yeonju Lee-Sikka, Peter Makarov, Aidan Malanoski, Sean Miller, Omar Ortiz, Reuben Raff, Arundhati Sengupta, Bora Seo, Yulia Spektor, and Winnie Yan. 2021. Results of the second sigmorphon shared task on multilingual grapheme-to-phoneme conversion. In *proc. The Seventeenth SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 115–125.

A.Vaswani, N.Shazeer, N.Parmar, J.Uszkoreit, L.Jones, A.N. Gomez, L. Kaiser, and I. Polosukhin. 2017. Attention is all you need. In *proc. The 31st Conference on Neural Information Processing Systems (NIPS 2017)*.

Shubham Bansal, Arijit Mukherjee, Sandeepkumar Satpal, and Rupeshkumar Mehta. 2020. On improving code mixed speech synthesis with mixlingual grapheme-to-phoneme model. In *proc. INTERSPEECH 2020*, pages 2957–2961.

Eunbi Choi, Hwa-Yeon Kim, Jong-Hwan Kim, and Jae-Min Kim. 2021. Label embedding for chinese grapheme-to-phoneme conversion. In *proc. INTERSPEECH 2021*.

Simon Clematide and Peter Makarov. 2021. Cluzh at sigmorphon 2021 shared task on multilin- gual grapheme-to-phoneme conversion: variations on a baseline. In *proc. The 18th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology.*

A. Deri and K. Knight. 2016. Grapheme-to-phoneme models for (almost) any language. In *proc. the 54th Annual Meeting of the Association for Computational Linguistics*.

Vasundhara Gautam, Wang Yau Li, Zafarullah Mahmood, Frederic Mailhot, Shreekantha Nadig, Riqiang Wang, and Nathan Zhang. 2021. Avengers, ensemble! benefits of ensembling in grapheme-to-phoneme prediction. In *proc. The 18th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology.*

Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *proc. The Sixth International Conference on Learning Representations*.

Jungo Kasai, Nikolaos Pappas, Hao Peng, James Cross, and Noah A. Smith. 2021. Deep encoder, shallow decoder: Reevaluating non-autoregressive machine translation. In *proc. The Ninth International Conference on Learning Representations*.

Hwa-Yeon Kim, Jong-Hwan Kim, and Jae-Min Kim. 2021. Nn-kog2p: A novel grapheme-to-phoneme model for korean language. In *proc. 2021 International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual bert? In *proc. the 57th Annual Meeting of the Association for Computational Linguistics*.

H. Sun, X. Tan, J. Gan, H. Liu, S. Zhao, T. Qin, and T. Liu. 2019a. Token-level ensemble distillation for grapheme-to-phoneme conversion. In *proc. INTERSPEECH 2019*.

Zhiqing Sun, Zhuohan Li, Haoqing Wang, Di He, Zi Lin, and Zhi-Hong Deng. 2019b. Fast structured decoding for sequence models. In *proc. The 33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, pages 2957–2961.

Kaili Vesik, Muhammad Abdul-Mageed, and Miikka Silfverberg. 2020. One model to pronounce them all: Multilingual grapheme-to-phoneme conversion with a transformer ensemble. In *proc. The 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 146–152.

Zijian Yang, Yingbo Gao, Weiyue Wang, and Hermann Ney. 2020. Predicting and using target length in neural machine translation. In *proc. the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*.

S. Yolchuyeva, G. Németh, and B. Gyires-Tóth. 2021. Transformer based grapheme-to-phoneme conversion. In *proc. INTERSPEECH 2019*.

Mingzhi Yu, Hieu Duy Nguyen, Alex Sokolov, Jack Lepird, Kanthashree Mysore Sathyendra, Samridhi Choudhary, Athanasios Mouchtaris, and Siegfried Kunzmann. 2020. Multilingual grapheme-to-phoneme conversion with byte representation. In *proc. ICASSP*, pages 33–40.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. In *proc. the 58th Annual Meeting of the Association for Computational Linguistics*.