INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Token-Level Ensemble Distillation for Grapheme-to-Phoneme Conversion

Hao Sun[1]*, Xu Tan*[2]*, Jun-Wei Gan*[3]*, Hongzhi Liu*[1]*, Sheng Zhao*[3]*, Tao Qin*[2] *and Tie-Yan Liu*[2]

[1]Peking University
[2]Microsoft Research
[3]Microsoft STC Asia

sigmeta@pku.edu.cn, xuta@microsoft.com, junwg@microsoft.com, liuhz@pku.edu.cn, Sheng.Zhao@microsoft.com, taoqin@microsoft.com, tyliu@microsoft.com

Abstract

Grapheme-to-phoneme (G2P) conversion is an important task in automatic speech recognition and text-to-speech systems. Recently, G2P conversion is viewed as a sequence to sequence task and modeled by RNN or CNN based encoder-decoder framework. However, previous works do not consider the practical issues when deploying G2P model in the production system, such as how to leverage additional unlabeled data to boost the accuracy, as well as reduce model size for online deployment. In this work, we propose token-level ensemble distillation for G2P conversion, which can (1) boost the accuracy by distilling the knowledge from additional unlabeled data, and (2) reduce the model size but maintain the high accuracy, both of which are very practical and helpful in the online production system. We use token-level knowledge distillation, which results in better accuracy than the sequence-level counterpart. What is more, we adopt the Transformer instead of RNN or CNN based models to further boost the accuracy of G2P conversion. Experiments on the publicly available CMUDict dataset and an internal English dataset demonstrate the effectiveness of our proposed method. Particularly, our method achieves 19.88% WER on CMUDict dataset, outperforming the previous works by more than 4.22% WER, and setting the new state-of-the-art results.

Index Terms: grapheme-to-phoneme conversion, knowledge distillation, transformer

1. Introduction

Grapheme-to-phoneme (G2P) conversion aims to generate a sequence of pronunciation symbols (phonemes) given a sequence of letters (graphemes), which is an important component in automatic speech recognition and text-to-speech systems [1, 2] to provide accurate pronunciations for the words not covered by the lexicon. G2P conversion can be viewed as a sequence to sequence task and modeled by the encoder-decoder framework. [3] adopt LSTM for G2P conversion and achieve improvements than the previous joint n-gram model [4]. [5] use convolutional sequence to sequence model and non-sequential decoding, and attain the previous best results on the public CMUDict dataset.

While previous works introduced the neural sequence to sequence models into G2P conversion and indeed achieved improvements over conventional methods, they did not take into account several practical issues of G2P conversion in the production system. First, considering training data is always costly through human labeling, how to further leverage the unlimited amount of unlabeled data is critical to improve the performance of G2P conversion. Second, large or ensemble models are too costly to serve when deploying in the online systems. How to reduce the model size but maintain high accuracy is essential.

Inspired by the knowledge distillation in computer vision [6, 7] and natural language processing [8, 9, 10], in this work, we propose the token-level ensemble distillation for G2P conversion, to address the practical problems mentioned above. First, we use knowledge distillation to leverage the large amount of unlabeled words. Specifically, we train a teacher model to generate the phoneme sequence as well as its probability distribution given unlabeled grapheme sequence, and regard the unlabeled grapheme sequence and the generated phoneme sequence as pseudo labeled data, and add them into the original training data. Second, we train a variety of models (CNN, RNN and Transformer) for ensemble to get higher accuracy, and transfer the knowledge of the ensemble models to a light-weight model that is suitable for online deployment, again by knowledge distillation. Besides, we adopt Transformer [11] instead of RNN or CNN as the basic encoder-decoder model structure, since it demonstrates advantages in a variety of sequence to sequence tasks, such as neural machine translation [11], text summarization [12], automatic speech recognition [13].

We conduct experiments on CMUDict 0.7b and our internal dataset, and also leverage additional unlabeled words crawled from the web. Our proposed method significantly boosts the accuracy of G2P conversion by 4.22% WER compared with the previous works. Specifically, Transformer model achieves higher accuracy than RNN and CNN based models, and token-level distillation outperforms sequence-level distillation.

Our contributions are listed as follows: (1) We propose token-level ensemble distillation for grapheme-to-phoneme conversion. (2) We are the first to use unlabeled words to boost the accuracy of grapheme-to-phoneme conversion, and also the first to introduce Transformer into this task and achieve better performance. (3) Our method achieves the state-of-the-art accuracy on CMUdict dataset, outperforming the previous best result by 4.22% WER.

2. Background

In this section, we briefly review the background of grapheme-to-phoneme conversion, Transformer model, as well as knowledge distillation.

2.1. Grapheme-to-Phoneme conversion

The G2P conversion is the process that generating the phoneme sequence (pronunciation) according to the grapheme sequence (word). G2P conversion is necessary and important as lexicon cannot cover all words, due to many words are long-tailed and a lot of new words and compound words appear. The spelling and pronunciation are not exactly corresponding for some languages, e.g. English. What is more, the alignments between graphemes and phonemes are complex. A grapheme may cor-

http://dx.doi.org/10.21437/Interspeech.2019-1208

respond to no phoneme, a single phoneme or many phonemes, as shown in Table 1, which makes G2P a hard task.

Table 1: *An example of the alignments between graphemes and phonemes.*

graphemes	B	U	B	B	L	E
phonemes	B	AH	null	B	AH:L	null

Joint sequence n-gram models have been widely used [4, 14, 15] for G2P conversion. Recently, sequence to sequence models have achieved great success in machine translation task [16, 17, 18, 19], and are soon applied on G2P conversion. [3] demonstrated that sequence to sequence models outperform joint sequence n-gram models. [20, 21] combined joint n-gram models with Bi-LSTM models and achieved good performance in G2P conversion. [5] adopted convolutional sequence to sequence model and proposed the non-sequential decoding [22] for G2P conversion, which achieved the previous state-of-the-art result on the public CMUDict 0.7b dataset.

While these sequence to sequence models achieve good performance on G2P conversion, there is still a gap when deploying online. In this work, we propose token-level ensemble distillation based on Transformer model, which can not only boost the accuracy of the G2P conversion with unlabeled words, but also reduce the model size for online deployment.

2.2. Transformer

Transformer [11] has achieved the state-of-the-art performance in many NLP tasks [23, 24, 25, 26]. The encoder and decoder in Transformer has N identical layers, and each layer in encoder consists of two different sub-layers: multi-head self-attention and feed-forward network, while the decoder has an additional multi-head attention sub-layer. Multi-head attention is to perform the attention function h times in parallel, allowing the model to jointly attend to information from different representation subspaces at different positions. Residual connection is employed between each sub-layer. Transformer can better model the interactions between any two tokens in the sequence and the computation of each token in the encoder and decoder can be parallel during training, which shows advantages over the RNN based models. To the best of our knowledge, this is the first work to apply Transformer in G2P conversion.

2.3. Knowledge Distillation

Knowledge distillation was first introduced by [27] for model compression, where a light student model can approximate the accuracy of a heavy and cumbersome teacher model. [6] first applied knowledge distillation on neural networks, and then a lot of works expand the usage of knowledge distillation to a variety of tasks, such as image classification [7, 28, 29] and natural language processing [8, 9, 10]. In this work, we leverage knowledge distillation to distill the knowledge from additional unlabeled word, as well as from the ensemble models, both of which are beneficial for the online production system.

3. Token-Level Ensemble Distillation

In this section, we propose the token-level ensemble knowledge distillation to boost the accuracy of G2P conversion, as well as reduce the model size for online deployment.

3.1. Token-Level Knowledge Distillation

Denote $D = \{(x, y) \in \mathcal{X} \times \mathcal{Y}\}$ as the training corpus which consists of the paired grapheme and phoneme sequence. A G2P model based on sequence to sequence learning aims to minimize the negative log-likelihood loss on corpus D:

$$\mathcal{L}_{NLL}(\theta) = -\sum_{(x,y)\in D} \log P(y|x;\theta), \tag{1}$$

where the likelihood $P(y \mid x; \theta)$ can be factored by the chain-rule and formulated as the cross-entropy between the one-hot label and per-token probability:

$$\log P(y|x;\theta) = \sum_{t=1}^{T_y}\sum_{k=1}^{|\mathcal{V}|} \mathbf{1}\{y_t = k\} \log P(y_t = k|y_{<t}, x; \theta), \tag{2}$$

where T_y is the length of the target sequence, $|\mathcal{V}|$ is the vocabulary size of the phonemes, y_t is the t-th target token in the phoneme sequence, and $\mathbf{1}\{\cdot\}$ is the indicator function indicating the id of the phoneme in vocabulary.

In token-level knowledge distillation, the one-hot label becomes the probability distribution output of the teacher model:

$$\begin{aligned}\mathcal{L}_{KD}(\theta) = -\sum_{(x,y)\in D}\sum_{t=1}^{T_y}\sum_{k=1}^{|\mathcal{V}|} Q(y_t = k|y_{<t}, x; \theta_T)\\ \times \log P(y_t = k|y_{<t}, x; \theta),\end{aligned} \tag{3}$$

where $Q(y_t = k|y_{<t}, x; \theta_T)$ is the probability distribution output of the teacher model θ_T.

3.2. Ensemble Distillation with Diverse Models

Model ensemble can incorporate the advantages of individual models, and reduce the effect of overfitting in a spirit of the bagging method [30]. However, the online production system cannot support large ensemble models for G2P conversion. Knowledge distillation is an effective way to distill the knowledge from strong ensemble models into single model. The ensemble distillation can be formulated as follows:

$$\begin{aligned}\mathcal{L}_{KD}(\theta) = -\sum_{(x,y)\in D}\sum_{t=1}^{T_y}\sum_{k=1}^{|\mathcal{V}|} \bar{Q}(y_t = k|y_{<t}, x)\\ \times \log P(y_t = k|y_{<t}, x; \theta),\end{aligned} \tag{4}$$

$$\bar{Q}(y_t = k|y_{<t}, x) = \frac{\sum_{m=1}^{M} Q(y_t = k|y_{<t}, x; \theta_T^m)}{M}, \tag{5}$$

where $\bar{Q}$ is the probability distribution combined by M models (θ_T^1 to θ_T^m), which is simply the average of the probability distribution of M models at each step of the target sequence.

The performance of the individual models and the diversity between them are essential for ensemble. On the one hand, we train deeper models to achieve higher accuracy. On the other hand, we choose Transformer [11], Bi-LSTM [18], and convolutional sequence to sequence [31] models to increase the diversity of ensemble models.

3.3. Knowledge Distillation with Unlabeled Source Words

In G2P conversion, it is easy to obtain abundant unlabeled source words (graphemes) from lexicon corpus of news or wikipedia. Knowledge distillation gives a way of using unlabeled source data. The teacher model can generate the target phoneme sequence given the unlabeled source grapheme sequence, and the generated phoneme sequence can be used as the label for student model. What is more, more unlabeled data can help distill the knowledge of the teacher model to the student model. In this work, we also use token-level knowledge distillation for unlabeled source words. Denote $D' = \{x \in \mathcal{X}\}$ as the corpus of unlabeled source words. The knowledge distillation loss with unlabeled source words is as follows:

$$\mathcal{L}'_{KD}(\theta) = -\sum_{x \in D'} \sum_{t=1}^{T_{y'}} \sum_{k=1}^{|\mathcal{V}|} \bar{Q}(y'_t = k|y'_{<t}, x) \times \log P(y'_t = k|y'_{<t}, x; \theta), \tag{6}$$

$$y' \sim \bar{Q}(y|x) \tag{7}$$

where y' is generated by the ensemble model (Equation 7), $Q(y'_t = k|y'_{<t}, x)$ is the probability distribution output of the ensemble model and is calculated by Equation 5.

The total loss of our method is the weighted combination of the original negative log-likelihood loss and the knowledge distillation loss [8, 10] on the labeled data, as well as the knowledge distillation loss on the unlabeled data:

$$\mathcal{L}_{TOTAL}(\theta) = (1-\lambda)\mathcal{L}_{NLL}(\theta) + \lambda\mathcal{L}_{KD}(\theta) + \mathcal{L}'_{KD}(\theta), \tag{8}$$

where each loss term is formulated in Equation 1, 4 and 6, λ is the weight to trade off between the two loss terms on labeled data.

4. Experiments and Results

In this section, we conduct experiments to verify the effectiveness of the proposed method. We first introduce the datasets used, and then describe the implementation details. At last, we report the results of our method and conduct some comparisons and analyses.

4.1. Experimental Setup

4.1.1. Datasets

We use two datasets to evaluate our proposed method: the first one is the publicly available CMUDict 0.7b and the other one is our internal dataset. For the public CMUDict 0.7b dataset, we use the same training/validation/test split (108952 training words, 5447 validation words and 12855 test words) as in [21], which is released in the CNTK toolkit[1]. The sizes of the grapheme and phoneme vocabulary are 27 and 39 respectively. To be consistent with the previous works [4, 5, 21], stress markings are removed and the multiple pronunciations are retained. Our internal dataset contains 184243 training words, 10837 validation words, 21678 test words, which includes uppercase and lowercase letters and stress markings. We keep the stress markings in training and ignore the stress during test. The sizes of the grapheme and phoneme vocabulary in our internal dataset are 54 and 73 respectively. We train our models on the training set and select the best hyperparameters according to the validation set.

We crawl nearly 2,000,000 unlabeled source words from the lexicon corpus of Google news[2]. As the crawled data contains words of other languages, unknown tokens and spelling errors, we first filter the data by removing the words with unknown tokens and then choose the top 300,000 unlabeled words according to their similarity to the training data[3].

4.1.2. Model Configurations

Ensemble Model We train the sequence to sequence based G2P models with different model structures for ensemble, including Transformer [11], Bi-LSTM [18] and CNN based sequence to sequence model [31]. We use 4 Transformer models, 3 CNN models and 3 Bi-LSTM models with different hyperparameters for ensemble, which give the best performance on the validation set. The 4 Transformer models share the same hidden size (256) but vary in the number of the encoder-decoder layers (6-6, 6-4, 8-6, 8-4). For the 3 CNN models, they share the same hidden size (256) but vary in the number of encoder-decoder layers (10-10, 10-10, 8-8) and convolutional kernel widths (3, 2, 2) respectively. For the 3 Bi-LSTM models, they share the same number of encoder-decoder layers (1-1), but with different hidden sizes (256, 384 and 512).

Student Model We choose Transformer as the student model and use the default configurations (256 hidden size and 6-6 layers of encoder-decoder) unless otherwise stated. We also vary the number of layers for the encoder and decoder to analyze and compare the accuracy and memory/time cost, which is essential for online deployment.

4.1.3. Training and Evaluation

We implement experiments with the fairseq-py[4] library in PyTorch. We use Adam optimizer for all models and follow the learning rate schedule in [11]. The dropout is 0.3 for Bi-LSTM and CNN models, while the residual dropout, attention dropout and ReLU dropout for Transformer models is 0.2, 0.4, 0.4 respectively. We set the λ in Equation 8 to 0.9 according to the validation performance. We train each model on 8 NVIDIA M40 GPUs. Each GPU contains roughly 4000 tokens in one mini-batch. We use beam search during inference and set beam size to 10. We use WER (word error rate) and PER (phoneme error rate) to measure the accuracy of G2P conversion. Edit distance is used in PER calculation. In WER calculation, considering the multiple pronunciations, word error is counted only when the output differs from all the references, following [4, 5, 21, 32].

4.2. Results and Analyses

4.2.1. Achieving State-Of-The-Art Accuracy

We first compare our method with previous works [4, 5, 21] on CMUDict 0.7b dataset, as shown in Table 2. Sequitur G2P [4] is a well established G2P conversion tool using joint sequence modelling and is widely used as a baseline for comparison. [21] used the ensemble of Bi-LSTM and joint n-gram model. The convolutional sequence to sequence model with non-sequential greedy decoding (NSGD) [5] is the previous state-of-the-art on CMUDict 0.7b dataset[5]. It can be seen that our method on 6-layer encoder and 6-layer decoder Transformer achieves

[1] https://github.com/Microsoft/CNTK/tree/master/Examples/SequenceToSequence/CMUDict/Data

[2] https://github.com/mmihaltz/word2vec-GoogleNews-vectors

[3] We use the distance between the 1/2/3-gram distribution of training words and unlabeled words, where the 1/2/3-gram means 1/2/3 consecutive characters.

[4] https://github.com/pytorch/fairseq

[5] They use a training/validation/test split different from [21] and ours. Therefore, we reproduce their work with on our training/ validation/test split, based on their public codebase (https://github.com/ctr4si/NSGD_G2P), and get similar result as theirs.

the new state-of-the-art result of 19.88% WER, outperforming NSGD by 4.22% WER.

Table 2: *Comparison between our method and the previous works on CMUDict 0.7b dataset.*

Method	PER	WER
Sequitur G2P [4]	6.12%	25.71%
Bi-LSTM + n-gram [21]	5.76%	24.88%
CNN with NSGD [5]	5.58%	24.10%
Our method	**4.60%**	**19.88%**

4.2.2. Reducing Model Size by 6x

Our method can also greatly reduce the model size for online deployment. We compare the WER, the number of parameters, and the inference speed between the baseline and our method, as shown in Table 3. The baseline method just uses transformer model (6-6 layers of encoder-decoder) without leveraging the ensemble knowledge distillation and unlabeled source words. To compare the inference speed, we use the time consumed by generating the outputs of the test set (12855 words) on a single M40 GPU with 12000 max tokens in one mini-batch. It can be seen from Table 3 that our method can still reach high accuracy with 1-1 layer of encoder-decoder, which can significantly reduce the model size by nearly 6 times and the time cost by nearly 4 times compared with the baseline model, but still achieving higher accuracy in terms of WER. The reduction in model size and inference time cost demonstrate the effectiveness of our method for online deployment.

Table 3: *Comparison of WER, number of parameters and inference time between the baseline and our method.*

Method	Layers	WER	Parameters	Time
Baseline	6-6	21.07%	11.09 millions	17.8s
Our method	1-1	20.25%	1.85 millions	4.4s

4.2.3. Analyses of Our Method

We first study the effect of distilling from unlabeled source words, as shown in Table 4. It can be seen that unlabeled source words can boost the accuracy by nearly 1% WER, demonstrating the effectiveness by introducing abundant unlabeled data into knowledge distillation.

Table 4: *Comparison of our method with and without unlabeled source words.*

Method	PER	WER
Without unlabeled data	4.78%	20.71%
With unlabeled data	4.60%	19.88%

We also compare token-level distillation with sequence-level distillation, where the student models are directly trained on the top-1 beam search results of the teacher network. As shown in Table 5, the result demonstrate the advantage of token-level distillation.

Furthermore, we study the effect of ensemble teacher model in knowledge distillation. As shown in Table 6, the ensemble teacher model can boost the accuracy by more than 1% WER, compared with the single teacher model (a Transformer model with 6-layer encoder and 6-layer decoder), which demonstrates the strong ensemble teacher model is essential to guarantee the performance of student model in knowledge distillation.

Table 5: *Comparison between token-level and sequence-level distillation.*

Method	PER	WER
Sequence-level	4.71%	20.32%
Token-level	4.60%	19.88%

Table 6: *Comparison of different teacher models for knowledge distillation.*

Method	PER	WER
Single teacher model	4.93%	21.05%
Ensemble teacher model	4.60%	19.88%

At last, we compare Transformer with RNN [21] and CNN [5] based models, without using knowledge distillation and unlabeled data, as shown in Table 7. We can see that Transformer model outperforms the RNN and CNN based models used in previous works, demonstrating the advantage of Transformer model.

Table 7: *Comparison of Transformer, LSTM and CNN.*

Method	PER	WER
Bi-LSTM + n-gram [21]	5.76%	24.88%
CNN with NSGD [5]	5.58%	24.10%
Transformer	4.96%	21.07%

4.2.4. Results on Our Internal Dataset

We compare our method with the previous state-of-the-art CNN with NSGD [5] (which is reproduced by ourself) on our internal dataset, as shown in Table 8. Our method outperforms CNN with NSGD by 3.52% WER, which demonstrates the effectiveness of our method for G2P conversion.

Table 8: *Results on our internal dataset.*

Method	PER	WER
CNN with NSGD [5]	3.79%	22.39%
Our method	**3.04%**	**18.87%**

5. Conclusion

In this work, we have proposed the token-level ensemble distillation with unlabeled source words for G2P conversion. Experiments on the publicly available CMUDict 0.7b dataset and our internal dataset demonstrate the effectiveness of our method on both improving the accuracy of G2P conversion and reducing the model size for online deployment. For future work, we will leverage more unlabeled data and pre-training [33] to improve the performance, and extend our work to other languages.

6. References

[1] Y. Ren, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Almost unsupervised text to speech and automatic speech recognition," in *ICML*, 2019.

[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," *arXiv preprint arXiv:1905.09263*, 2019.

[3] K. Yao and G. Zweig, "Sequence-to-sequence neural net models for grapheme-to-phoneme conversion," *arXiv preprint arXiv:1506.00196*, 2015.

[4] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech communication*, vol. 50, no. 5, pp. 434–451, 2008.

[5] M.-j. Chae, K. Park, L. Bang, S. Suh, L. Park, N. Kimt, and L. Park, "Convolutional sequence to sequence model with non-sequential greedy decoding for grapheme to phoneme conversion," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 2486–2490.

[6] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[7] T. Furlanello, Z. C. Lipton, M. Tschannen, L. Itti, and A. Anandkumar, "Born again neural networks," *arXiv preprint arXiv:1805.04770*, 2018.

[8] Y. Kim and A. M. Rush, "Sequence-level knowledge distillation," *arXiv preprint arXiv:1606.07947*, 2016.

[9] M. Freitag, Y. Al-Onaizan, and B. Sankaran, "Ensemble distillation for neural machine translation," *arXiv preprint arXiv:1702.01802*, 2017.

[10] X. Tan, Y. Ren, D. He, T. Qin, and T.-Y. Liu, "Multilingual neural machine translation with knowledge distillation," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=S1gUsoR9YX

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[12] C. Gong, X. Tan, D. He, and T. Qin, "Sentence-wise smooth regularization for sequence to sequence learning," 2018.

[13] S. Zhou, L. Dong, S. Xu, and B. Xu, "A comparison of modeling units in sequence-to-sequence speech recognition with the transformer on mandarin chinese," in *International Conference on Neural Information Processing*. Springer, 2018, pp. 210–220.

[14] S. F. Chen, "Conditional and joint models for grapheme-to-phoneme conversion," in *Eighth European Conference on Speech Communication and Technology*, 2003.

[15] K. Wu, C. Allauzen, K. Hall, M. Riley, and B. Roark, "Encoding linear models as weighted finite-state transducers," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[16] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[17] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1412–1421.

[18] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.

[19] H. Hassan, A. Aue, C. Chen, V. Chowdhary, J. Clark, C. Federmann, X. Huang, M. Junczys-Dowmunt, W. Lewis, M. Li *et al.*, "Achieving human parity on automatic chinese to english news translation," *arXiv preprint arXiv:1803.05567*, 2018.

[20] S. Toshniwal and K. Livescu, "Jointly learning to align and convert graphemes to phonemes with neural attention models," in *2016 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2016, pp. 76–82.

[21] B. Milde, C. Schmidt, and J. Köhler, "Multitask sequence-to-sequence models for grapheme-to-phoneme conversion." in *INTERSPEECH*, 2017, pp. 2536–2540.

[22] J. Guo, X. Tan, D. He, T. Qin, L. Xu, and T.-Y. Liu, "Non-autoregressive neural machine translation with enhanced decoder input," in *AAAI*, 2019.

[23] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[24] M. Dehghani, S. Gouws, O. Vinyals, J. Uszkoreit, and Ł. Kaiser, "Universal transformers," *arXiv preprint arXiv:1807.03819*, 2018.

[25] A. Wangperawong, "Attending to mathematical language with transformers," *arXiv preprint arXiv:1812.02825*, 2018.

[26] D. R. So, C. Liang, and Q. V. Le, "The evolved transformer," *arXiv preprint arXiv:1901.11117*, 2019.

[27] C. Bucilu, R. Caruana, and A. Niculescu-Mizil, "Model compression," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2006, pp. 535–541.

[28] R. Anil, G. Pereyra, A. Passos, R. Ormandi, G. E. Dahl, and G. E. Hinton, "Large scale distributed neural network training through online distillation," *arXiv preprint arXiv:1804.03235*, 2018.

[29] C. Yang, L. Xie, S. Qiao, and A. Yuille, "Knowledge distillation in generations: More tolerant teachers educate better students," *arXiv preprint arXiv:1805.05551*, 2018.

[30] T. G. Dietterich, "Ensemble methods in machine learning," in *International workshop on multiple classifier systems*. Springer, 2000, pp. 1–15.

[31] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 1243–1252.

[32] K. Rao, F. Peng, H. Sak, and F. Beaufays, "Grapheme-to-phoneme conversion using long short-term memory recurrent neural networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4225–4229.

[33] K. Song, X. Tan, T. Qin, J. Lu, and T.-Y. Liu, "Mass: Masked sequence to sequence pre-training for language generation," in *International Conference on Machine Learning*, 2019, pp. 5926–5936.