A UNIFIED FRONT-END FRAMEWORK FOR ENGLISH TEXT-TO-SPEECH SYNTHESIS

Zelin Ying, Chen Li, Yu Dong, Qiuqiang Kong, YuanYuan Huo, Yuping Wang, Yuxuan Wang

Speech, **A**udio & **M**usic **I**ntelligence (SAMI), ByteDance

yingzelin@bytedance.com

Abstract

The front-end is a critical component of English text-to-speech (TTS) systems, responsible for extracting linguistic features that are essential for a text-to-speech model to synthesize speech, such as prosodies and phonemes. The English TTS front-end typically consists of a text normalization (TN) module, a prosody word prosody phrase (PWPP) module, and a grapheme-to-phoneme (G2P) module. However, current research on the English TTS front-end focuses solely on individual modules, neglecting the interdependence between them and resulting in sub-optimal performance for each module. Therefore, this paper proposes a unified front-end framework that captures the dependencies among the English TTS front-end modules. Extensive experiments have demonstrated that the proposed method achieves state-of-the-art (SOTA) performance in all modules.

Index Terms: text-to-speech front-end, text normalization, prosody word prosody phrase, grapheme-to-phoneme

1. Introduction

Speech synthesis is a critical technology that has brought significant convenience to the lives of people and has been extensively utilized in various scenarios, including voice assistants [1, 2] and audiobooks [3, 4]. In English text-to-speech (TTS) synthesis, the front-end plays a vital role by extracting various linguistic features from raw text to provide the acoustic model with sufficient information for synthesizing natural speech. Enhancing the accuracy and efficiency of the front-end will lead to improved quality and naturalness of the TTS synthesized speech.

The English TTS front-end is typically composed of three linguistic-related modules, including: 1) A text normalization (TN) module, which converts non-standard words such as numbers, symbols, and abbreviations into spoken-form words. 2) A prosody word prosody phrase (PWPP) module, which identifies pause boundaries for different duration levels in the sentence. 3) A grapheme-to-phoneme (G2P) module, which converts word sequences into phoneme sequences. For instance, the word "hello" is converted to "HH EH L OW".

Researchers have focused on developing innovative approaches for each module. For the TN module, Ebden et al. [5] proposed a rule-based method using weighted finite-state transducers (WFST) to tokenize the input and convert the classified tokens based on their semiotic class. Due to the limited coverage of rules, Sproat et al. [6] proposed different recurrent neural network (RNN) architectures to learn the correct normalization function. Zhang et al. [7] proposed a hybrid method that combines rules and models to integrate the advantages of a rule-based model and a neural model for text normalization. For the PWPP module, a simple approach is to add pauses after punctuation marks to indicate prosody. However, recent studies have used sequence tagging neural network models [8, 9] to predict prosody. Regarding the G2P module, the most common method is to find common words in the lexicon and use neural network models [10, 11] to predict out-of-vocabulary (OOV) words. Additionally, the part-of-speech (POS) task was proposed to differentiate between homographs, which are written the same but pronounced differently depending on their part-of-speech context.

Although there exist unified front-end frameworks in other languages [12, 13], they cannot be compared due to the language-specific differences. In the English TTS front-end, previous studies have mostly focused on improving individual modules and achieved promising results. However, some issues still persist: 1) No systematic English TTS front-end solution was proposed. 2) Those linguistic-related modules may promote each other, separating them may cause the whole English TTS front-end sub-optimal. 3) In the G2P module, some homographs with the same part-of-speech need to be solved. Therefore, it becomes particularly imperative and urgent to propose a unified English TTS front-end framework considers the interplay among the different modules to achieve improved performance.

In this paper, we propose a novel unified front-end framework for English TTS that integrates the TN, PWPP, and G2P modules into a cohesive system to tackle the above challenges. The major contributions of our research are summarized as follows:

- We propose a systematic English TTS front-end framework that employs a shared multi-task model to address English TTS front-end tasks. To the best of our knowledge, this is the first work to unify English TTS front-end tasks.
- Our approach optimizes each module of the front-end. Specifically, the TN module achieves more accurate categorization and offers greater flexibility and control in correcting bad cases. The hierarchical PWPP method better utilizes containing relationships between prosody labels. And the G2P module introduces a Polyphone task to improve the accuracy of homographs with the same part-of-speech.
- We conduct extensive experiments in each module and achieve state-of-the-art (SOTA) results in all of them, which sheds light on the path towards a unified multi-task model in the English TTS front-end.

2. Methodology

In this section, we provide an overview of our proposed framework, followed by a detailed description of the workflow of the TN, PWPP, and G2P modules within the framework.

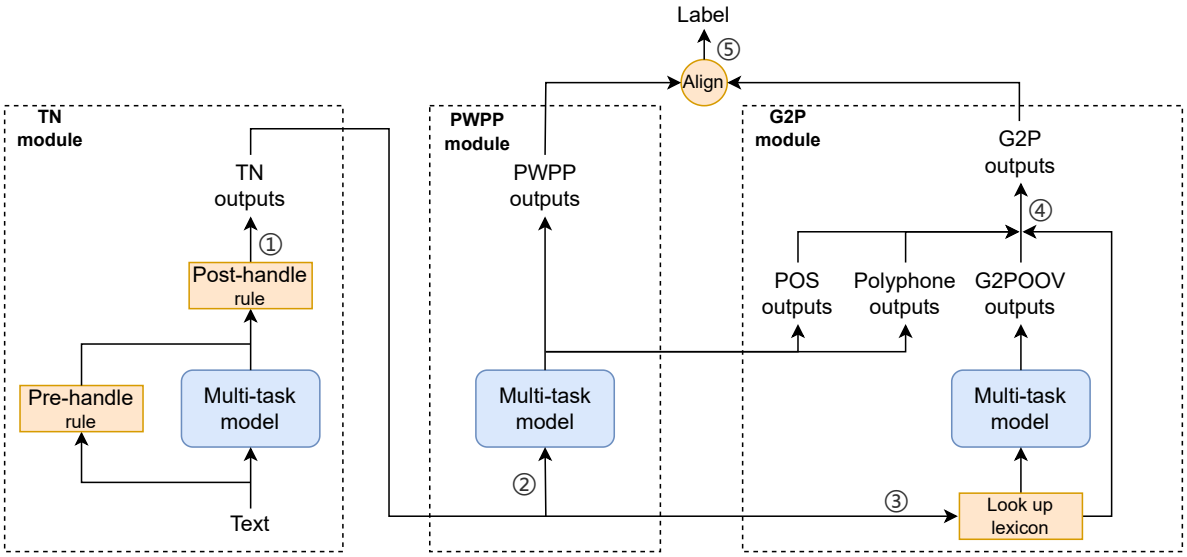

Figure 1: *Flowchart of our proposed framework*

2.1. Framework Overview

Figure 1 illustrates the overall flowchart of our proposed framework, which includes the TN, PWPP, and G2P modules. Our framework is composed of a shared multi-task model and well-designed rules. The multi-task model supports five tasks: TN, PWPP, G2POOV, POS, and Polyphone. The TN and PWPP tasks serve their respective modules, while the G2POOV, POS, and Polyphone tasks serve the G2P module.

The dataflow in our proposed framework is as follows: ① We first input raw text to the TN module, which applies a hybrid method combining rules and models to predict the TN outputs; ② Then, we feed the TN outputs into the shared multi-task model to predict the PWPP, the POS, and the Polyphone outputs; ③ Next, we look up the TN outputs words in the lexicon and directly obtain the pronunciations for the found words, otherwise refer to the G2POOV outputs for the OOV words; ④ Then, we combine phoneme results of the lexicon and OOV words and update homographs pronunciations by the POS and the Polyphone outputs to acquire G2P outputs; ⑤ Finally, we align PWPP outputs and G2P outputs to obtain the normative label, in which each row shows a phoneme-level feature result.

2.2. Modules

2.2.1. TN Module

The TN module converts non-standard words, such as numbers, symbols, and abbreviations, into spoken-form words. It includes a multi-task model and an expert-designed rule system. Given the input text, we use the model to predict one of 19 corresponding categories and apply pre-handle rules to correct any misclassified categories. The data for these corrected categories is then fed into post-handle rules to transcribe and obtain the TN transcription results. For the TN sequence tagging task (TN task) in the multi-task model, we define 19 categories, including "CARDINAL", "DIGIT", etc. The category "O" indicates "other" and does not need to be normalized. Each normalized category is accompanied by the beginning, inside, end, and single (BIES) positional tags indicating word boundaries information. We combine the categories, their word boundaries, and the "O" category to form $19 \times 4 + 1 = 77$ labels for sequence tagging. The architecture of the TN task model, shown in Figure 2, consists of a BERT [14] model, a 1-D **C**onvolution **B**ank and bidirectional **G**RU (CBG) [15] encoder, and a masked **C**onditional **R**andom **F**ields (CRF) [16] layer. We use a CRF loss to train the TN task. Compare to other English TN studies such as [6] and [17], which only divide text into 16 categories and may not cover all cases accurately, our approach addresses such limitations. For example, in previous studies, both "dvd" and "dvds" are classified into the "LETTERS" category, resulting in the incorrect transcription of "dvds" as "d v d s" instead of "d v ds". To overcome such issues, we introduce three new categories, namely "LETTERSS", "MATH", and "SCORE", to the existing 16 categories, thereby covering plural letter sequences, mathematical formulas, and scores. Using more detailed categories allows the model to learn with greater precision and achieve better results. Besides, the previous approach relies too heavily on the model and places too much trust in its ability, resulting in difficulties controlling the final output, such as whether to transcribe "911" as "nine hundred and eleven" or "nine one one". The model frequently struggles in these situations. Our hybrid approach, which utilizes both rules and models, effectively handles these issues and is also flexible enough to accurately transcribe hot words.

2.2.2. PWPP Module

The PWPP module identifies pause boundaries for different duration levels in sentences, relying on the PWPP sequence tagging task (PWPP task) outputs of the multi-task model. In the PWPP task, pause duration time is divided into three levels, ranging from low to high: prosody word level #1, prosody phrase level #2, and intonation phrase level #3. Instead of directly tagging the three prosody levels, a hierarchical sequence tagging structure is employed to independently tag each prosody level. Specifically, each prosody level is transformed into a binary classification task to predict the probability of the corresponding prosody levels after the words. Finally, the PWPP outputs are determined by adopting the highest predicted prosody level. Figure 2 shows that the model structure of each prosody level is composed of a BERT model, followed by a CBG encoder and a CRF layer. We use a CRF loss to train the binary classification task of each prosody level and sum all prosody level losses to obtain the PWPP task loss.

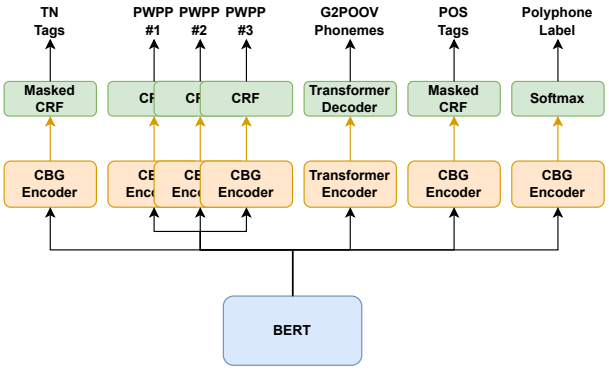

Figure 2: *Multi-task model structure*

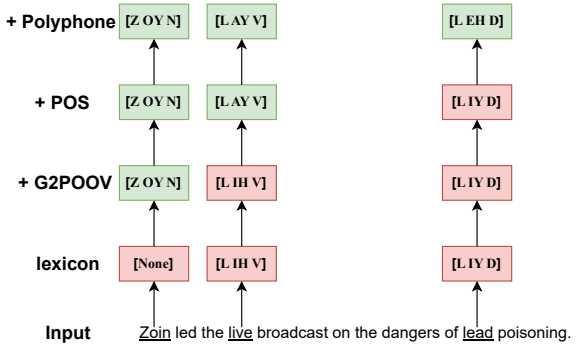

Figure 3: *G2P module workflow, red blocks mean wrong phonemes and green blocks mean correct phonemes*

2.2.3. G2P Module

The G2P module converts word sequences into phoneme sequences, which depends on the lexicon and the G2POOV task to achieve candidate pronunciations of each word. Then, we employ the POS and the Polyphone tasks to select the pronunciations of homographs. The G2POOV sequence to sequence task (G2POOV task) is used to generate phoneme sequences for the OOV words. We define 61 characters, including letters, numbers, and punctuation. We expect the model to convert the input characters to output phoneme sequences of 73 phonemes. As shown in Figure 2, the model architecture of the G2POOV task is composed of a BERT model, followed by a Transformer [18] encoder and a Transformer decoder. The input to the G2POOV task is an OOV word such as "Zoin", which will be tokenized by character level to obtain "Z o i n" sequence to output phoneme sequences such as "Z OY N". We use a cross entropy loss to train the G2POOV task. Although previous studies, such as [19], have used Transformer-based models to solve the G2POOV task, our approach differs in that we utilize a BERT model that has been fine-tuned on multiple tasks and employ beam search with size k during inference to consider multiple candidate solutions. This allows our method to better utilize global information and avoid local optimum, leading to improved performance. The POS sequence tagging task (POS task) aims to distinguish homographs pronunciations with different part-of-speech. We define 24 part-of-speech categories for the POS task, such as "Noun" and "Verb". Similar to the TN module, we combine categories, their BIES word boundaries, and the "O" category to form 97 labels for sequence tagging. The model architecture of the POS task is similar to the TN task and is shown in Figure 2. We use a CRF loss to train the POS task. The Polyphone sequence classification task (Polyphone task) mainly optimizes homographs with different pronunciations in the same part-of-speech depending on context. We ordered the possible pronunciations of each word as classification labels. For example, the noun "lead" pronounced "L IY D" for one possible meaning and "L EH D" for another. The model architecture of the Polyphone task is composed of a BERT model, followed by a CBG encoder and a softmax classifier, as shown in Figure 2. The loss for training the Polyphone task is a cross entropy loss. To sum up, the workflow of the G2P module is shown in Figure 3. Common words will obtain pronunciations from the lexicon, OOV words will obtain pronunciations from the G2POOV task, and homographs pronunciations will be updated based on the POS and Polyphone tasks, to obtain the G2P outputs.

3. Experimental Evaluations

3.1. Experimental Settings

We conduct extensive experiments in the TN, PWPP, and G2P modules. For evaluating the TN module, we use the Google open source English text normalization dataset [17] that includes 755,441 sentences. We compare our method with the SOTA method Seq2Edits [20] and other recently proposed methods, such as RNN-based [21] and Transformer-based [22] methods. We use the sentence error rate (SER) [22] as the evaluation metric, which measures the proportion of incorrect sentences to the total number of sentences. A sentence is considered incorrect if the predicted output does not exactly match the reference. For the evaluation of the PWPP module, we use an internal dataset including 100,026 sentences. We compare our proposed hierarchical sequence tagging method, which independently predicts each prosody level, against the traditional sequence tagging method, which utilizes the BERT-CBG-CRF structure to directly predict the tags of three prosody levels. We adopt sequence tagging F1-score as the evaluation metric. In the G2P module, we train and validate the POS task using 102,164 sentences, and the Polyphone task using 28,097 sentences, while we utilize a dictionary to train and validate the G2POOV task. Our G2POOV approach is compared with the SOTA method r-G2P [23], as well as other recently proposed methods, including Transformer-based G2P [19] and encoder-decoder with global attention [24], to evaluate performance. In addition, we gather a real-world test set including 2560 sentences to evaluate the overall performance of the complete G2P module. Moreover, we carry out experiments with various beam search sizes to demonstrate the effect of this technique. We also conduct ablation experiments to determine the individual contributions of the G2POOV, POS, and Polyphone tasks within the G2P module. Our evaluation metric is the word error rate (WER) [25], where a word is deemed incorrect if the predicted output does not match the reference exactly. WER is defined as the ratio of the total number of incorrectly predicted words to the total number of words in the evaluated text.

In our framework, we use a pre-trained multilingual BERT [14] model with 12 layers as the language model. We set the hidden unit of the CBG encoder to 256, and the model dimension of the Transformer to 256 with 4 heads in multi-head attention between the Transformer encoder and decoder. Additionally, we set the beam search size to 3 and train the Transformer decoder using teacher forcing. The model learning rate is set to 5×10^{-5}, and we use AdamW [26] as the optimizer.

Table 1: *SERs in the TN module test set*

Methods	SER (%)
RNN-based [22]	1.80
Transformer-based [21]	1.42
Seq2Edits [20]	1.36
BERT-CBG-CRF (ours)	**1.19**

Table 2: *Tagging F1-scores in the PWPP module test set*

Methods	Prosody level	F1-score (%)
Traditional seq. tagging	#1	61.15
	#2	56.37
	#3	82.63
Hierarchical seq. tagging (ours)	#1	**90.83**
	#2	**57.65**
	#3	**83.36**

3.2. Experimental Results and Analysis

We first evaluate the SER in the TN module, compared with other methods. Table 1 shows that the RNN-based model in [22] achieves an SER of 1.80%, proving RNN-based model cannot achieve good enough results due to the limited encoding ability. Transformer-based model in [21] achieves an SER of 1.42%, due to the improved encoding ability of the Transformer encoder and the help of the BERT pre-trained language model. The Seq2Edits approach in [20] treats TN as a sequence of edit operations, uses span-level edits to capture compact local representations and achieves an SER of 1.36%. Our proposed method uses a BERT model to enhance word embeddings, and a CBG encoder that considers both local and global text features, plus the help of well-designed rules, therefore achieves the best SER of 1.19%, outperforms the SOTA method (Seq2Edits) by 0.17% SER.

Then, we assess the tagging F1-score of different prosody levels in the PWPP module. As presented in Table 2, our proposed hierarchical sequence tagging method outperforms the traditional sequence tagging method. More specifically, the hierarchical sequence tagging method shows a significant improvement over the traditional sequence tagging method in the #1 prosody level and a slight improvement in the #2 and #3 prosody levels. It is due to the hierarchical sequence tagging method considers the constraint that high level prosody belongs to the low level prosody, resulting in better overall performance.

Next, we evaluate the G2POOV task and the entire G2P module. Table 3 presents the results of various methods for the G2POOV task. The vanilla Transformer [19] fails to achieve good performance, with a WER of 22.10%. [24] uses an encoder-decoder architecture with global attention to capture more global information and improve performance of the model, achieving a WER of 21.69%. [23] introduces three methods for controllable noise to improve robustness of the model, achieving a WER of 19.85%. Our proposed method, based on the BERT model and leveraging several front-end linguistic-related tasks, as well as beam search to obtain the globally optimal solution, achieves the best performance WER of 19.42%. Furthermore, we compare the effects of different beam search sizes on the final model performance. Without beam search, the model performs the worst, as it selects a local optimal solution, resulting in poor performance. As the beam search size increases to 3, the performance of the model gradually improves. However, larger beam search sizes do not further improve the performance of the model and instead increase computation costs, which is not worth it. In the complete G2P module, we evaluate our system on a real-world test set of 2560 sentences. As shown in Table 4, using a lexicon to search for word pronunciations results in a WER of 3.83%. One reason for this is that the lexicon may contain incorrect entries. Another reason is that out-of-vocabulary (OOV) words and homographs cannot be handled. To address this, we add the G2POOV task to the G2P module, enabling the system to handle OOV words, which reduce the WER to 3.42%. Next, we add the POS task to handle homographs that are pronounced differently depending on their part-of-speech, resulting in a WER decrease to 3.17%. Finally, we add the Polyphone task to handle homographs that are pronounced differently even within the same part-of-speech, resulting in a WER decrease to 3.09%. These results demonstrate the importance of the G2POOV, POS, and Polyphone tasks in the G2P module. To better illustrate the influence of the Polyphone task, we take some homographs with the same part-of-speech as case studies, the accuracy of "used", "lead", and "tear" increased from 56.91% to 83.74%, from 55.55% to 80.95%, and from 53.33% to 93.33%, respectively. Since then, our proposed framework consistently achieves the best performance in all TN, PWPP, and G2P evaluations, and the effectiveness is strongly demonstrated.

Table 3: *WERs in the G2POOV task on CMUDict test set*

Methods	WER (%)
Transformer 4x4 [19]	22.10
Encoder-decoder + global attention [24]	21.69
r-G2P [23]	19.85
BERT-Transformer w/o beam search (ours)	19.49
BERT-Transformer w/ beam size 2 (ours)	19.46
BERT-Transformer w/ beam size 3 (ours)	**19.42**
BERT-Transformer w/ beam size 5 (ours)	19.42
BERT-Transformer w/ beam size 10 (ours)	19.42

Table 4: *WERs in the G2P module real-world sentences*

Methods	WER (%)
lexicon	3.83
lexicon + G2POOV	3.42
lexicon + G2POOV + POS	3.17
lexicon + G2POOV + POS + Polyphone	**3.09**

4. Conclusion

In this paper, we propose a unified front-end framework for English text-to-speech synthesis to solve complicated front-end tasks, including TN, PWPP, and G2P modules. Our method achieves SOTA performance in all modules, with a sentence error rate of 1.19% in the TN module, F1-scores of 90.83%, 57.65%, and 83.36% for the #1, #2, and #3 prosody levels in the PWPP module, a word error rate of 19.42% in the G2POOV task, and a sentence error rate of 3.09% in the complete G2P module. To the best of our knowledge, this is the first work to unify English front-end tasks. Our proposed method may inspire more research work in the literature and cast a major impact on the front-end for English text-to-speech synthesis.

5. References

[1] S. Subhash, P. N. Srivatsa, S. Siddesh, A. Ullas, and B. Santhosh, "Artificial intelligence-based voice assistant," in *IEEE World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*, 2020, pp. 593–596.

[2] S. Shalini, T. Levins, E. L. Robinson, K. Lane, G. Park, and M. Skubic, "Development and comparison of customized voice-assistant systems for independent living older adults," in *International Conference on Human-Computer Interaction*. Springer, 2019, pp. 464–479.

[3] A. Bérard, L. Besacier, A. C. Kocabiyikoglu, and O. Pietquin, "End-to-end automatic speech translation of audiobooks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 6224–6228.

[4] K. J. Esteves and E. Whitten, "Assisted reading with digital audiobooks for students with reading disabilities," *Reading Horizons*, vol. 51, no. 1, p. 21, 2011.

[5] P. Ebden and R. Sproat, "The Kestrel TTS text normalization system," *Natural Language Engineering*, vol. 21, no. 3, pp. 333–353, 2015.

[6] R. Sproat and N. Jaitly, "An RNN model of text normalization," *Proc. Interspeech 2017*, pp. 754–758, 2017.

[7] J. Zhang, J. Pan, X. Yin, C. Li, S. Liu, Y. Zhang, Y. Wang, and Z. Ma, "A hybrid text normalization system using multi-head self-attention for mandarin," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6694–6698.

[8] Y. Zheng, J. Tao, Z. Wen, and Y. Li, "BLSTM-CRF based end-to-end prosodic boundary prediction with context sensitive embeddings in a text-to-speech front-end." in *Interspeech*, 2018, pp. 47–51.

[9] Y. Yan, J. Jiang, and H. Yang, "Mandarin prosody boundary prediction based on sequence-to-sequence model," in *IEEE Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, vol. 1, 2020, pp. 1013–1017.

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

[11] I. Illina, D. Fohr, and D. Jouvet, "Grapheme-to-phoneme conversion using conditional random fields," in *Annual Conference of the International Speech Communication Association*, 2011.

[12] J. Pan, X. Yin, Z. Zhang, S. Liu, Y. Zhang, Z. Ma, and Y. Wang, "A unified sequence-to-sequence front-end model for mandarin text-to-speech synthesis," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6689–6693.

[13] Y. Zhang, L. Deng, and Y. Wang, "Unified mandarin tts front-end based on distilled bert model," *arXiv preprint arXiv:2012.15404*, 2020.

[14] J. D. M.-W. C. Kenton and L. K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of NAACL-HLT*, 2019, pp. 4171–4186.

[15] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio *et al.*, "Tacotron: Towards end-to-end speech synthesis," *Proc. Interspeech 2017*, pp. 4006–4010, 2017.

[16] T. Wei, J. Qi, S. He, and S. Sun, "Masked conditional random fields for sequence labeling," in *North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021, pp. 2024–2035.

[17] R. Sproat and N. Jaitly, "RNN approaches to text normalization: A challenge," *arXiv preprint arXiv:1611.00068*, 2016.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[19] S. Yolchuyeva, G. Németh, and B. Gyires-Tóth, "Transformer based grapheme-to-phoneme conversion," *Proc. Interspeech 2019*, pp. 2095–2099, 2019.

[20] F. Stahlberg and S. Kumar, "Seq2Edits: Sequence transduction using span-level edit operations," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 5147–5159.

[21] J. H. Ro, F. Stahlberg, K. Wu, and S. Kumar, "Transformer-based models of text normalization for speech applications," *arXiv preprint arXiv:2202.00153*, 2022.

[22] H. Zhang, R. Sproat, A. H. Ng, F. Stahlberg, X. Peng, K. Gorman, and B. Roark, "Neural models of text normalization for speech applications," *Computational Linguistics*, vol. 45, no. 2, pp. 293–337, 2019.

[23] C. Zhao, J. Wang, X. Qu, H. Wang, and J. Xiao, "r-g2p: Evaluating and enhancing robustness of grapheme to phoneme conversion by controlled noise introducing and contextual information incorporation," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 6197–6201.

[24] 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.

[25] S. F. Chen *et al.*, "Conditional and joint models for grapheme-to-phoneme conversion." in *INTERSPEECH*. Citeseer, 2003.

[26] I. Loshchilov and F. Hutter, "Fixing weight decay regularization in adam, iclr 2018 conference," in *International Conference on Learning Representations (ICLR)*, 2018.