IMPROVING PHONETIC REALIZATIONS IN TTS BY USING PHONEME-ALIGNED GRAPHEMES

Manish Sharma[⋆]*, Yizhi Hong*[†]*, Emily Kaplan*[†]*, Siamak Tazari*[†]*, Rob Clark*[⋆]

Google UK[⋆], Google USA[†]

ABSTRACT

Most text-to-speech acoustic models, such as WaveNet, Tacotron, ClariNet, etc., use either a phoneme sequence *or* a letter sequence as the fundamental unit of speech. Although the letter (or grapheme) sequence closely matches the actual runtime input of the TTS system, it often fails to represent the fine-grained phonetic variations. A purely phonemic input seems to perform better in practice, though is heavily dependent on a meticulously crafted phonology and lexicon. This reliance poses issues (with quality and consistency) which can lead to the need for a trade-off between quality and scalability. To overcome this, we propose using a mix of the two inputs, namely providing phoneme-aligned graphemes to the model. In this paper, we show that this approach can help the model learn to disambiguate some of the more subtle phonemic variations (such as the realization of reduced vowels), and that this effect improves the fidelity to the accent of the original voice talent. For evaluation, we present a way of generating an unbiased targeted test using phoneme spectral diffs, and using that, show improvement over the baseline approach for multiple voice technologies and multiple locales.

Index Terms— Graphemes, Phonology, Schwa, Vowels, Accent

1. INTRODUCTION

With recent neural network architectures like WaveNet [1], Tacotron [2], WaveGlow [3] and others [4], text-to-speech (TTS) technology is now on the cusp of human-level performance. In fact, some recent works have claimed to achieve human-level performance, characterized by equal preference for synthesized versus natural speech [5]. However, these analyses often fail to highlight some of the more subtle, but consistent, flaws in these models. In this work, we will discuss our approach to address one such shortcoming.

Although there have been works [6, 7, 8, 9] that employ purely orthographic input for training TTS, recent acoustic models [10, 11, 12] mostly use a phonemic sequence as input to the neural networks. This is because phonemes are often a better representation of the spoken-form, as compared to the orthography. The reliance on a purely phonemic input leads to dependency on the accuracy and completeness of the following:

1. **Phonology** An exhaustive set of permissible phonemes, and associated rules for a language.
2. **Lexicon** - A database of mappings from words to possible phoneme sequences representing their pronunciations.
3. **G2P** (Grapheme-to-phoneme) An algorithm for mapping words not covered by the lexicon to phoneme sequences.

Although phonology definition and lexicon curation may seem trivial, they force the developer to take a prescriptivist approach to the development of a language. Namely, they must define some target variety (and its pronunciation conventions) and consistently apply them to a lexicon meant to serve the language in its entirety (and thus also its potential multitude of varieties). In other words, they must choose between a narrow and a broad transcription system [13]. While ideally, a narrow phonetic transcription system would be the most suitable input for TTS, it is not easily maintainable or scalable. In practice, this often leads to the enforcement of many-to-one mappings, e.g. reducing most unstressed vowels to the schwa phoneme [14], in an attempt to better scale lexicon development. This coarse phonology (and thereby lexicon) compounds with the phone-level audio to create a layer of abstraction between the two inputs that the models see. At training time, the text-phoneme pairs are often algorithmically determined using, for example, an HMM-based aligner [15]. This aligner uses phonology rules and learnt probability matrices to select the best-match variants of a word from the lexicon. If that lexicon is under-specified or too far abstracted from the speech-signal this can also lead to under-specification (or over-generalization) in the resulting synthesis.

In our work, we propose to use graphemes as an *addtional* input feature when training a neural TTS model, and show that it helps improve the realizations of several phonemes in a linguistic context. A grapheme sequence is a (possibly empty) substring of letters, present in the written-form of the text. Our motivations to include graphemes are three-fold. Firstly, we hope to maintain the benefits of using a more abstract phonemic inventory, namely that it prevents large-scale divergence in the lexicon and that it helps us scale. Secondly, it is believed that a phonology and lexicon which are more under-specified than over-specified are better suited to the modelling of multiple different accents within a given language (since the speech-signal-to-phoneme relationship is already fairly abstract). If the acoustic model is better equipped to learn the relationships between speech-signal and phonemes, we propose that this in turn enhances the fidelity of the synthesized accent to the speaker. Finally, we expect that the additional grapheme signal will help compensate for occasional errors in the lexicon, e.g. those bound to arise over several years of development by many contributors.

At a higher level, graphemes are closer to the meaning and intent of the text whereas phonemes are closer to the speech. Many languages exhibit a high level of *logography* implying that reconstructing the graphemes from the phonemes is difficult [16]. Thus, omitting graphemic information is akin to withholding crucial information from the models. However, this aspect can be compensated by adding word-embeddings (see Section 2), and therefore, the focus of our current work is on low-level pronunciation differences.

In order to evaluate the efficacy of our proposed approach, we present a way to automatically build a targeted test set using spectral differences between the corresponding phones in the synthesized audio samples (Sec 4.1). The improvements that result from adding the phoneme-aligned graphemes are subtle, but they are consistent across locales and voice technologies. For a certain German accent, we also highlight the pronunciation differences using first and second formants in the speech.

2. RELATED WORK

The problem of aligning graphemes-to-phonemes for use in TTS is not new. The majority of the previous research focuses on using grapheme-to-phoneme (or G2P, or letter-to-sound, or LTS) alignment for a scalable lexicon development. This is crucial for any TTS system, because it is practically impossible to have a lexicon with all possible words and their pronunciations in a language. One popular approach to achieving this alignment is defining a superset of grapheme-to-phoneme mappings for a language, followed by maximizing likelihoods using approaches like Expectation-Maximization [17], Classification and Regression trees [18] or other custom decision trees [19, 20]. Alternatively, [21, 22] have also employed neural networks to achieve G2P, albeit, without explicitly generating the alignments. Our work is different from G2P because we use the grapheme identities as a linguistic feature in our acoustic models, not for lexicon development.

Furthermore, there have been works that propose to use graphemic information implicitly in a TTS system using word embeddings. For example, [5, 23, 24] used BERT embeddings [25] to improve the perceived semantics of a TTS system. Similarly, [26, 27] use other forms of word embeddings to augment TTS. Our work explicitly uses graphemes as linguistic features, without any preprocessing, thereby, helping the model focus more on robust pronunciations than semantics. [28] presented a representation mixing approach to improve pronunciations. While they propose a new training methodology, ours is a feature augmentation approach. Moreover, our analysis focuses extensively on highlighting pronunciation differences using our targeted test set generation approach.

3. SYSTEM OVERVIEW

In this section, we describe our experimental setup, the methodology to align phonemes and graphemes, and finally augment these graphemes to the phonemic input while training a text-to-speech model.

3.1. Preparing training input

Each word in the recorded utterances is added to a pronunciation lexicon, offering one or more pronunciation variants. Additionally, in order to train our acoustic model, we need to annotate the phonemes and align the associated audio segments for each. Since this is non-trivial to achieve manually, we use an HMM-based aligner [15] that achieves both - selecting the most suitable pronunciation variant for an utterance from the lexicon, and aligning these phonemes to the correct part of the recording. Finally, we compute linguistic features at different levels of the sentence hierarchy [29] and concatenate them at phoneme level to construct our baseline TTS training input.

3.2. Phoneme and grapheme alignment

In order to add the graphemic features, we need to align the graphemes to the phonemes, which is not a well-defined task. Figure 1 shows multiple types of mappings that can occur between graphemes and phonemes in English. Furthermore, there can be multiple valid alignments for a single word. For example, consider the word "cause" with pronunciation /k A z/; we could choose to align se $\rightarrow$ /z/, or alternatively, align s $\rightarrow$ /z/ and e $\rightarrow$ ϕ. Because of these ambiguities, we aim for a "best-match" alignment that attempts to be consistent in its decisions.

Most lexicons do not usually provide phoneme-grapheme alignments (unlike Combilex [18]), therefore these must build their own aligner. There are two popular ways in which we can build such an aligner: using a neural network, or using a finite state transducer (FST) [30]. While both approaches can produce coherent results, an FST-based alignner is more interpretable, and hence we describe and use it for our experiments and results.

For an FST-based phoneme-grapheme aligner, we first create a collection of permissible phoneme-grapheme pairs, or *graphones*. This collection of graphones will represent the nodes in our FST. In order to generate this collection, we first create a large pool of all permissible mappings. These can be hand-written and expanded semi-automatically by looking for words in the lexicon that don't align, and adding missing mappings. Then we build a minimal subset as follows: starting with an empty collection of graphone mappings, we greedily add mappings from the pool until we have minimized our alignment metrics on a list of most common words for the language. The alignment metrics are such that we aim to:

1. Maximize the number of successful word alignments
2. Minimize graphemes aligned to no phoneme
3. Minimize phonemes aligned to no grapheme

Moreover, we applied a lower cost for mapping vowel graphemes to vowel phonemes, so that these are preferred. Finally, for inferring alignments on a query word and its phoneme sequence, we find the least cost path in this FST. The grapheme identities in the graphone pair are provided as an augmented linguistic feature to the model.

3.3. Neural acoustic models

In this work, our proposal is to augment the input feature space, without changing anything in the acoustic models. However, in order to extensively assess the implications of this grapheme augmentation, we experiment with a number of popular voice technologies. These are briefly discussed below:

1. **WaveRNN**: This is an autoregressive RNN based acoustic model [11]. Our slightly modified version of WaveRNN uses a conditioning stack comprising of dilated convolutions and a sampling stack comprising of a Gated Recurrent Unit (GRU) that upsamples the feature stream, and projects onto an audio signal.
2. **WaveNet**: The WaveNet acoustic model trains in two steps. First, we train an autoregressive network [1], and then distill an inverse autoregressive flow-based model from it, popularly known as Parallel WaveNet [10]. In our experiments, we use the same training configuration of this network as described in [10].
3. **CHiVE+Vocaine**: Vocaine [31] is a lightweight universal vocoder that reconstructs audio from a set of vocoder features: mel cepstrum, aperiodicity, fundamental frequency (F0) and voicing. As an intermediate network, we use a modified version of Clockwork Hierarchical Variational Autoencoder (CHiVE) [12] to predict these vocoder features from the input linguistic features. Unlike previous two acoustic models, CHiVE+Vocaine is more suited for low-end or embedded TTS technologies.

Note that the aforementioned networks require input features at frame-level. Here, a frame is defined as the smallest segment of audio that determines the resolution of phoneme durations and pitch tracking [32]. These frame-level features are generated by a prosody model. In our work, we use the CHiVE-BERT model [24] for synthesizing prosody at runtime. Note that, the choice of the prosody model is not crucial to our analysis because the pronunciation variations are mostly agnostic of the applied prosody, assuming the source-filter theory of speech synthesis [33].

Fig. 1. Multiple possible ways of aligning graphemes with phonemes

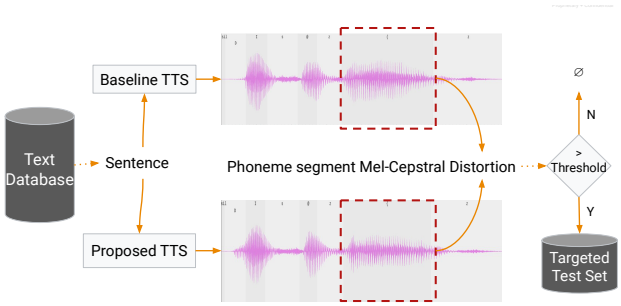

Fig. 2. Targeted test set for side-by-side evaluation

4. EXPERIMENTS

Using the methodology described above, we train a baseline model, and our proposed model (i.e. using an additional phoneme-aligned grapheme feature) and compare the two. We run our experiments on four locales: American-English, British-English, Indian-English and German. Finally, to analyze the efficacy of the grapheme feature, we employ the following objective evaluation metrics

4.1. Side-by-side on targeted test sets

Our experimentation is motivated by the harnessing of supplementary orthographic information to enrich the pronunciation and phonology. In order to capture these subtle changes, we outline a process to automatically generate a targeted test set, as shown in Figure 2. We use a large text corpus to synthesize audio samples using the baseline and proposed voice pipelines. Specifically, we used a dataset selection methodology as described in [34] to optimize the diphone and triphone coverage of the source dataset. This source dataset (shown in dark on the left in Fig 2) is then used to synthesize audio samples using the baseline and our proposed voice. For each sentence, we compute the Mel Cepstral Distortion (MCD) cost between the audio segments corresponding to the same phonemes. Since the two segments might have different durations, we use a Dynamic Time Warping (DTW) distance. Finally, sentences containing phonemes that produce a high MCD-DTW are collated into our targeted test set. Our hypothesis is that the phonemes in these sentences will sound different.

4.2. Side-by-side on standard test sets

This test set is composed of a set of 1000 queries to a TTS system from sources like assistant conversations, driving directions, translation and search results. This test set is similar to the one used in [1, 11, 12]. The objective of this test set is to quantify holistic changes in the voice quality.

For the German voice, we employ formant analysis to test for accent fidelity, instead of a side-by-side test (Section 5.3). Our side-by-side evaluations receive at most 12 ratings from a rater. Each evaluation item presents two audio samples to the rater synthesized from the baseline and our proposed voice. For each such stimulus, the rater provides a rating in $\{-3, -2, -1, 0, 1, 2, 3\}$ depending on which side they prefer, which are then combined into the test result. We use student's t-test [35] to determine the statistical significance of our test result, choosing a threshold of $p < 0.01$.

Table 1. Results of side-by-side on targeted test set

	WaveRNN	WaveNet	CHiVE-Vocaine
British English	**0.573 ± 0.149** p=0.00	**0.392 ± 0.122** p=0.00	**0.290 ± 0.102** p=0.00
Indian English	**0.244 ± 0.109** p=0.00	**0.109 ± 0.081** p=0.00	**0.124 ± 0.109** p=0.00
American English	**0.223 ± 0.130** p=0.00	**0.154 ± 0.166** p=0.01	**0.056 ± 0.100** p=0.01

Table 2. Results of side-by-side on standard test set

	WaveRNN	WaveNet	CHiVE-Vocaine
British English	**0.017 ± 0.018** p=0.01	0.015 ± 0.018 p=0.03	0.012 ± 0.015 p=0.03
Indian English	-0.008 ± 0.016 p=0.20	0.005 ± 0.018 p=0.42	-0.001 ± 0.019 p=0.92
American English	0.005 ± 0.021 p=0.55	-0.003 ± 0.018 p=0.74	0.002 ± 0.020 p=0.80

5. RESULTS AND DISCUSSION

The results of side-by-side evaluations, as described in Sec 4.2 and 4.1 are presented in table 2 and 1 respectively. Side-by-side results are represented by the mean of ratings and the associated 95% confidence interval. A positive rating means that the rater preferred the grapheme voice. Results that are statistically significant (i.e. p-value of two sided t-test < 0.01) are highlighted in bold.

In table 1, we find consistent wins for all voice technologies across all the locales on the targeted test set. We will elucidate the key phenomena we observed in these side-by-side tests in the following subsections [1]. From Table 2, we do not see a substantial improvement in the overall voice quality. However, this result importantly shows that we do not inadvertently degrade the overall voice quality. Even though WaveRNN on en-GB seems to be doing overall better on the standard test set, we do not see similar performance across all voice technologies. Note that the confidence intervals in

[1] Audio samples are available at `https://google.github.io/phoneme-aligned-graphemes-in-tts`

table 1 are bigger than in table 2, because the test set size is 10 times larger for the standard test, as compared to the targeted set.

5.1. Fine grained nuances in phonology

TTS is a one-to-many problem. Even a single phoneme can have multiple realizations in the phonetic space. Consider the sentence for a British English speaker: "Rosa's Roses" [36, 37]. In this sentence, both "Roses" and "Rosa's" are broadly transcribed /r\@Uz@z/ (in X-SAMPA notation). However, for many speakers, the pronunciation of schwa is subtly contrastive for these two words. Specifically, the schwa in "Rosa's" is a true central schwa vowel, but "Roses" is realized with a close-central vowel (barred-i or ɨ) [36, 38], i.e. /r\@Uz@z/ versus /r\@Uz1z/. While contrastive for many speakers of English, this distinction is not universal, and even in those with the distinction, the distribution of barred-i (and other such reduced vowels) varies greatly from speaker to speaker. Injecting the grapheme alignment to a single reduced schwa vowel provides an additional cue to the models that helps it to achieve the correct realization of the vowel according to the original speaker's distribution.

5.2. Resilience to transcription inconsistencies

As discussed earlier, phonology definition is a hard problem, as is lexicon creation. The aligner used for variant selection might not always produce the intended alignments. Similarly, during inference, the G2P or lexicon variant selection might not always be accurate, or there may be variation in what should be systematic handling (which pollutes the signal). In all such cases, it is useful to have an additional phoneme-aligned grapheme feature that helps guide the acoustic model to the intended phonetic realization.

One such example occurs in the words "is" and "as" in US-English, where both have a pronunciation variant that can be transcribed as /@ z/ (with a generic schwa). Now consider a sentence where they both occur in a row, such as "This is as confusing". If the text normalization frontend decides to use the /@ z/ pronunciation for both, the model will have no knowledge of the original difference in those words. However, with our approach, the original graphemes, "i" and "a", are aligned with the schwas, respectively, and thus the model is able to produce the subtle but important difference in coloring those schwas accordingly. Indeed, this was preferred by the raters consistently across several examples in our evaluation.

5.3. Accent fidelity

A more accurate realization in the phonetic space also entails a higher degree of accent fidelity to the original speaker, i.e. the accent of the synthesized voice more closely resembling the accent of the speaker. In our experiments, we observed that phoneme-aligned grapheme features were more efficient in matching the pronunciations with the accent of the speaker.

For German in particular, we considered two broad accents: one spoken in the northern regions and the other spoken in the southern regions. One popular example of the accent disambiguation is that in the south, the words containing "ä" are pronounced with /E/ and not /e/, that is, using a short open-mid vowel instead of the long close-mid one. We observed this accent in some of our speakers in several high-frequency words like: tätig, ähnlichen, beschädigt, allmählich, etc. In our experiments, the training dataset for German consisted of 8 speakers, out of which 2 could be classified as having this accent. Additionally, the lexicon contained only the northern Germanic pronunciation variants for most words. In our analysis,

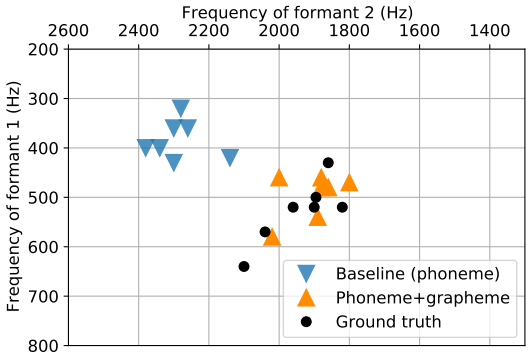

Fig. 3. Plots of first two formants F1 and F2 of the vowel corresponding to ä in German words found from targeted test set - tätig, ähnlichen, schwäbischen, beschädigt, allmählich, qualität, fakultäten. Formants in the grapheme approach are closer to the accent of the speaker, denoted by formants found in recordings, represented by the black dots.

we found that the baseline TTS did not successfully propagate the accent of southern speakers onto the synthesized speech. However, since the phoneme-aligned grapheme had access to text information, it was more faithful in reproducing the source speaker's accent. Specifically, for southern speakers with recordings of words containing "ä", our approach was able to learn to rely more on the grapheme (ä), than the overgeneralized input phoneme /e/. Note that, in our side-by-side tests, the raters simply chose the correct version of the pronunciation according to their bias. Hence, to present the result more meaningfully, we ommitted side-by-side results, and instead show the plot of the first and second formants of vowels in some of the words, and compare these to the formants of the words found in recordings of a speaker. Figure 3 shows that the grapheme voice is closer to the speaker's pronunciation than the baseline approach.

While we noticed many improvements in our voices, we did notice a few pronunciation degradations because of the model relying excessively on the grapheme component. For example, for a male Indian English speaker in the sentence "Turn right onto Holder road", 'Holder' was pronounced with a close-mid vowel instead of the mid-central schwa. Nevertheless, the degradations did not follow a strict pattern, and were outnumbered by the improvements, as also evident from the results on the targeted test set in Table 1.

6. CONCLUSION

We discussed an approach that uses phoneme-aligned graphemes as an additional input to training acoustic models in TTS. We showed that this approach can be used to improve phonetic realizations using a broad phonology definition. Using our automatically-generated targeted test set, we showed that our approach outperforms the baseline, on several locales, and the improvements generalize across popular voice technologies like WaveNet and WaveRNN. The augmentation of graphemes can help the models learn fine grained nuances in pronunciations, for example with accents, and also makes the model robust to lexical imperfections. This provides a scalable way of dealing with large lexicons in TTS.

7. ACKNOWLEDGEMENT

The authors would like to thank the wider Google TTS team for their help with the experimentation and quality feedback.

8. REFERENCES

[1] A. v. d.Oord, S.Dieleman, H.Zen, K.Simonyan, O.Vinyals, A.Graves, N.Kalchbrenner, A.Senior, and K.Kavukcuoglu, "WaveNet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[2] J.Shen, R.Pang, R. J.Weiss, M.Schuster, N.Jaitly, Z.Yang, Z.Chen, Y.Zhang, Y.Wang, R.Skerrv-Ryan, et al., "Natural TTS synthesis by conditioning wavenet on Mel spectrogram predictions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4779–4783.

[3] R.Prenger, R.Valle, and B.Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 3617–3621.

[4] X.Tan, T.Qin, F.Soong, and T.-Y.Liu, "A survey on neural speech synthesis," *arXiv preprint arXiv:2106.15561*, 2021.

[5] Y.Jia, H.Zen, J.Shen, Y.Zhang, and Y.Wu, "PnG BERT: Augmented BERT on phonemes and graphemes for neural TTS," *arXiv preprint arXiv:2103.15060*, 2021.

[6] 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," *arXiv preprint arXiv:1703.10135*, 2017.

[7] W.Ping, K.Peng, and J.Chen, "Clarinet: Parallel wave generation in end-to-end text-to-speech," *arXiv preprint arXiv:1807.07281*, 2018.

[8] J.Sotelo, S.Mehri, K.Kumar, J. F.Santos, K.Kastner, A.Courville, and Y.Bengio, "Char2wav: End-to-end speech synthesis," 2017.

[9] W.Wang, S.Xu, B.Xu, et al., "First step towards end-to-end parametric TTS synthesis: Generating spectral parameters with neural attention.," in *INTERSPEECH*, 2016, pp. 2243–2247.

[10] A.Oord, Y.Li, I.Babuschkin, K.Simonyan, O.Vinyals, K.Kavukcuoglu, G.Driessche, E.Lockhart, L.Cobo, F.Stimberg, et al., "Parallel wavenet: Fast high-fidelity speech synthesis," in *International conference on machine learning*. PMLR, 2018, pp. 3918–3926.

[11] N.Kalchbrenner, E.Elsen, K.Simonyan, S.Noury, N.Casagrande, E.Lockhart, F.Stimberg, A.Oord, S.Dieleman, and K.Kavukcuoglu, "Efficient neural audio synthesis," in *International Conference on Machine Learning*. PMLR, 2018, pp. 2410–2419.

[12] T.Kenter, V.Wan, C.-A.Chan, R.Clark, and J.Vit, "CHiVE: Varying prosody in speech synthesis with a linguistically driven dynamic hierarchical conditional variational network," in *International Conference on Machine Learning*. PMLR, 2019, pp. 3331–3340.

[13] J.Laver and L.John, *Principles of phonetics*, Cambridge university press, 1994.

[14] E.Flemming, "The phonetics of schwa vowels," *Phonological weakness in English*, pp. 78–98, 2009.

[15] D.Talkin and C. W.Wightman, "The aligner: Text to speech alignment using markov models and a pronunciation dictionary," in *The Second ESCA/IEEE Workshop on Speech Synthesis*, 1994.

[16] R.Sproat and A.Gutkin, "The taxonomy of writing systems: How to measure how logographic a system is," *Computational Linguistics*, pp. 1–54, 2021.

[17] R. I.Damper, Y.Marchand, J.-D.Marseters, and A.Bazin, "Aligning letters and phonemes for speech synthesis," in *Fifth ISCA Workshop on Speech Synthesis*, 2004.

[18] K.Richmond, R. A.Clark, and S.Fitt, "Robust LTS rules with the combilex speech technology lexicon," in *INTERSPEECH*, 2009.

[19] J.Basu, T.Basu, M.Mitra, and S. K. D.Mandal, "Grapheme to phoneme (G2P) conversion for bangla," in *2009 Oriental COCOSDA International Conference on Speech Database and Assessments*. IEEE, 2009, pp. 66–71.

[20] D.Braga, L.Coelho, and F. G. V.Resende, "A rule-based grapheme-to-phone converter for TTS systems in European Portuguese," in *2006 International Telecommunications Symposium*. IEEE, 2006, pp. 328–333.

[21] 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.

[22] B.Peters, J.Dehdari, and J.van Genabith, "Massively multilingual neural grapheme-to-phoneme conversion," *arXiv preprint arXiv:1708.01464*, 2017.

[23] Y.Xiao, L.He, H.Ming, and F. K.Soong, "Improving prosody with linguistic and BERT derived features in multi-speaker based Mandarin Chinese neural TTS," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6704–6708.

[24] T.Kenter, M. K.Sharma, and R.Clark, "Improving Prosody of RNN-based English Text-To-Speech Synthesis by Incorporating a BERT model," in *INTERSPEECH*, 2020.

[25] 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.

[26] P.Wang, Y.Qian, F. K.Soong, L.He, and H.Zhao, "Word embedding for recurrent neural network based TTS synthesis," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4879–4883.

[27] X.Wang, S.Takaki, and J.Yamagishi, "Enhance the word vector with prosodic information for the recurrent neural network based TTS system.," in *INTERSPEECH*, 2016, pp. 2856–2860.

[28] K.Kastner, J. F.Santos, Y.Bengio, and A.Courville, "Representation mixing for tts synthesis," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5906–5910.

[29] H.Zen, "An example of context-dependent label format for HMM-based speech synthesis in english," *The HTS CMUARCTIC demo*, vol. 133, 2006.

[30] M.Mohri, "Finite-state transducers in language and speech processing," *Computational linguistics*, vol. 23, no. 2, pp. 269–311, 1997.

[31] Y.Agiomyrgiannakis, "Vocaine the vocoder and applications in speech synthesis," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4230–4234.

[32] D.Talkin and W. B.Kleijn, "A robust algorithm for pitch tracking (RAPT)," *Speech coding and synthesis*, vol. 495, pp. 518, 1995.

[33] J.Harrington, S.Cassidy, and S.Cassidy, *Techniques in speech acoustics*, vol. 8, Springer Science & Business Media, 1999.

[34] M.Podsiadlo and V.Ungureanu, "Experiments with training corpora for statistical text-to-speech systems," in *INTERSPEECH*, 2018.

[35] T. K.Kim, "T test as a parametric statistic," *Korean journal of anesthesiology*, vol. 68, no. 6, pp. 540, 2015.

[36] E.Flemming and S.Johnson, "Rosa's roses: Reduced vowels in American English," *Journal of the International Phonetic Association*, vol. 37, no. 1, pp. 83–96, 2007.

[37] P.Ladefoged and S. F.Disner, *Vowels and consonants*, John Wiley & Sons, 2012.

[38] F. B.Parkinson, *The representation of vowel height in phonology*, The Ohio State University, 1996.