INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

Enhancing Sequence-to-Sequence Text-to-Speech with Morphology

Jason Taylor, Korin Richmond

The Centre for Speech Technology Research, The University of Edinburgh, UK
jason.taylor@ed.ac.uk, korin@cstr.ed.ac.uk

Abstract

Neural sequence-to-sequence (S2S) modelling encodes a single, unified representation for each input sequence. When used for text-to-speech synthesis (TTS), such representations must embed ambiguities between English spelling and pronunciation. For example, in *pothole* and *there* the character sequence *th* sounds different. This can be problematic when predicting pronunciation directly from letters. We posit pronunciation becomes easier to predict when letters are grouped into subword units like morphemes (e.g. a boundary lies between *t* and *h* in *pothole* but not *there*). Moreover, morphological boundaries can reduce the total number of, and increase the counts of, seen unit subsequences. Accordingly, we test here the effect of augmenting input sequences of letters with morphological boundaries. We find morphological boundaries substantially lower the Word and Phone Error Rates (WER and PER) for a Bi-LSTM performing G2P on one hand, and also increase the naturalness scores of Tacotrons performing TTS in a MUSHRA listening test on the other. The improvements to TTS quality are such that grapheme input augmented with morphological boundaries outperforms phone input without boundaries. Since morphological segmentation may be predicted with high accuracy, we highlight this simple pre-processing step has important potential for S2S modelling in TTS.

Index Terms: Speech Synthesis, Sequence-to-Sequence, Morphology, Pronunciation

1. Introduction

The English spelling system is notoriously confusing. Take the word *coa**th**a**ng**er*. It contains the digraphs *th* and *ng* which have different pronunciations in words like ***th**ere* or *ra**ng**e*. The default solution in text-to-speech (TTS) for such issues is a complex pipeline of processing known collectively as the front-end.

The front-end's main function is to predict the phone sequence corresponding to input text. At the minimum, it typically exploits a lexicon containing phonetic transcriptions. These provide phone sequences for in-vocabulary words, and training data for a grapheme-to-phoneme (G2P) model that predicts pronunciations for out-of-vocabulary (OOV) words. Example TTS lexica include CMUdict [1], Unisyn [2] and Combilex [3].

A comprehensive front-end will also expand abbreviations (e.g. *AAPL* or *AMZN*), and disambiguate non-standard words (like numbers) and homographs. Examples of comprehensive front-end packages for TTS include Festival [4], Mary [5] and Sparrowhawk [6].

Front-ends were originally developed to work hand-in-hand with monotonic acoustic models when acoustics for each segment were only predicted frame by frame, from left to right. In this approach, text with ambiguous pronunciation needed phones as an intermediate representation. This was because the pronunciation of any character could not be dependent on characters at a long distance down in a sequence.

Yet sequence-to-sequence (S2S) modelling has enabled acoustic prediction directly from text in end-to-end (E2E) TTS (e.g. Tacotron [7]) which is now often considered state-of-the-art in acoustic modelling [8]. By encoding the entirety of each input sequence at once, acoustics for any single character are predicted according to that character's wider environment. This means the complexity in English spelling is learnt as an implicit letter-to-sound model. In other words, phonetic input is no longer necessary for acoustic modelling in languages with an unclear mapping between letters and sounds. In addition, E2E TTS development is simple thanks to a single, monolithic model that jointly performs pronunciation, acoustic and duration modelling. As this approach gains traction in the TTS field (c.f. the ESPnet-TTS project [9]), the continuing role of any separate front-end processing is naturally brought in question. Given its cost in time and money to develop, what value does it bring, and how can it be simplified and optimised for S2S TTS?

Mulitple reports have in fact been emerging in the literature showing that S2S TTS incorporating linguistic features can give superior performance. Indeed, state-of-the-art quality has only so far been achieved for English with phonetic input [10]. Syntactic features improved intonation in [11]. Using clockwork RNNs to hierarchically represent multiple levels of linguistic features (syllabic, phonetic and lexical) with a variational autoencoder also improved intonation in [12]. Language modelling features were found to speed-up convergence during training in [13]. We do not refer to these systems as E2E since the additional features require intervention to produce.

We posit that further benefits to S2S TTS may be gained from using morphology, which has the advantage of being relatively easy to derive, as supplementary information. At its simplest, morphology can delineate meaningful sub-word units. For instance, *hanger* is composed of the root *hang* and bound morpheme *er*. These may attach to the root *coat* to derive *coathanger*. Morpheme boundaries can thus in principle help resolve some of the pronunciation confusion arising from English spelling: the implications of the sequence *th* in {*coat*}{*hang*}>*er*> are clearer than in *coathanger*.

Even more significantly, morphological information has the potential to increase the utility of the available training data. In the field of neural machine translation it has been demonstrated that breaking down text input into morphemes both lessens the total translatable vocabulary and increases the frequency of each (now shorter) vocabulary item, which ultimately boosts overall performance [14]. For instance, the count of *coat* is increased when the compound *coathanger* is split. We argue breaking down character sequences into shorter, common subsequences in this way is likely to bring analogous benefits for TTS model training. To demonstrate how morpheme boundaries reduce the total number of recurring character subsequences, and also increase their frequencies, in Figure 1 we compare the words and morphemes present in the LJ speech dataset [15]. Note how the morphemes (derived from Unisyn) occur with higher counts

http://dx.doi.org/10.21437/Interspeech.2020-1547

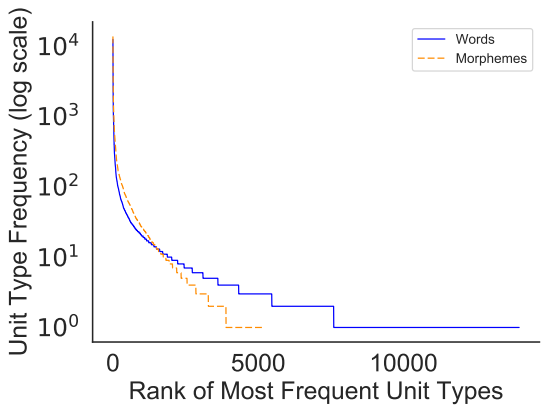

Figure 1: *Total counts of most frequent units: words and morphemes in the training data for our TTS systems. Splitting into morphemes reduces the vocabulary and increases the counts of seen units.*

than the original words, as shown by the area where the morpheme (dotted) line is above the words (blue) line. As the V column in Table 1 shows, across the entire dataset the vocabulary is more than halved thanks to morphological boundaries.

Morphological information comes included with high quality lexica such as Unisyn, but a practical TTS system using morphology would of course have to handle OOVs too. Automatic morphological decomposition could be employed for these, and indeed it has been shown a model can predict the location and identity of morpheme boundaries with reasonably high accuracy (e.g. [16]). Though this is an important future consideration, we focus at this stage on assessing the potential benefit of morphological features for S2S TTS generally, assuming they are freely available. We thus restrict experiments to in-vocabulary words to achieve our aims here.

In addition to evaluating how the inclusion of morpheme boundaries affects speech synthesised by a full S2S TTS model, we also investigate them in this paper in the context of S2S G2P models. Since G2P models typically have simpler architectures and fewer trainable parameters, they serve as less compute-heavy proxies to understand how a full S2S TTS model may deal with pronunciations. As such, we effectively consider the G2P models as performing a similar pronunciation modelling task, though predicting explicit phone strings rather than the "latent" pronunciations in S2S TTS, which are only inferrable from the synthesised audio. As well as lower computational cost, they also offer the benefit of a definite discrete output representation that can be easily scored against a target phone string, so providing convenient additional insight.

In summary, our aim is to demonstrate the benefits of morphological features for S2S-based G2P and TTS. In the rest of this paper, we first demonstrate that morphological features improve S2S G2P performance by lowering Word and Phone Error Rates (WER and PER). We then present results from an evaluation using an open-source Tacotron model trained multiple times with and without morphological features. We assess the performance of these S2S TTS models in a MUSHRA listening test, and show that including morpheme boundaries can indeed improve naturalness scores.

Table 1: *Description of the various types of input fed to Tacotron. V is the total vocabulary size, i.e. number of unique units (words or morphs), comprised of graphemes or phones.*

Input	Base Unit	Format	V
G	Graphemes	p o t h o l e s	13981
GM	Graphemes	{ p o t } { h o l e } >s >	5202
P	Phones	p o t h ou l z	12631
PM	Phones	{ p o t } {h ou l } >z >	5606

2. Experimental Setup

2.1. Morphology

Though it does not feature in many lexicons (e.g. CMUdict), morphological composition is indicated for all words contained in both Unisyn and Combilex. Unisyn provides surface-level morphemes where entries are interspersed directly with boundaries. For instance, the word *unanswered* has prefix <*un*–, root {*answer*}, and suffix –*ed*>. The entry with both letters and morphemes appears as <*un*<{*answer*}>*ed*> in Unisyn notation. Unisyn is freely downloadable for academic purposes and also contains phones, syllable boundaries, lexical stress markers and POS tags. Its morphology is very simple to predict. While this enables a relatively easy application to out-of-vocabulary (OOV) words, a more detailed and fine-grained notation could potentially add further benefit. For instance, canonical morphology [17] modifies each detected unit to one of a standardised set. Take *acquirability*: its surface representation in Unisyn is <*a*{*cquir*}>*abil* >*ity* >, but canonical segments would be more consistent:<{*acquire*}>*able*>*ity* >, and thus increase the frequency even more of the morphemes curve in Figure 1. At this stage, we are not analysing how to optimise morphological representations for S2S TTS, but are rather just exploring the viability of using morphology in S2S-TTS models at all. For convenience in this paper, we therefore simply use the morphology provided in Unisyn as-is. Further information on the approach taken to morphology in Unisyn is provided in [18].

2.2. G2P Models

We first evaluated the effect of morphological boundaries to augment input to a neural S2S G2P model. We used the base-form lexicon of Unisyn, designed to be accent-independent, which contains 160,000 entries split into a ratio of 75:20:5 for the training, validation and test sets. Prior to training our models, We created two partitions of the data for *random* and *disjoint* test sets. For the *random* test set, we randomly selected 20% and 5% of Unisyn entries to be included in the val and test sets respectively. For the *disjoint* test set, entries were grouped according to the primary root morpheme of the words, and the validation and test sets were selected such that they contained distinct sets of root morphemes. For example, the root {hiccough} may have been in the training set with associated entries such as {hiccough}>ed>and {hiccough}>ing>, but {cough} is a separate root morpheme and could therefore be put instead in the test set with derivations such as {cough}>ed>and {cough}>ing>. Note, the sets were made entirely disjoint in terms of root morphemes from one another. In this way, we test the G2P model's ability to generalise to unseen root morphemes, as often happens in real applications of G2P. No further pre-processing steps were taken, such as removing homographs, apostrophes or words with fewer than 4

letters.

Although the transformer network may have been shown to perform with the lowest WER when using CMUdict [19], for computing ease we trained BiLSTMs, another neural S2S model competitive in terms of WER [20]. We use the OpenNMT package built on Pytorch [21], using 3 bi-directional encoder and decoder layers with 500 units each, a learning rate of 0.001, and Luong's global attention [22] with dropout of 0.1. The networks were trained with mini-batches of 64 and optimised with ADAM. For the results reported here, the BiLSTM converged after 50,000 training steps.

2.3. TTS Systems

We trained TTS models on a subset of the Linda-Johnson speech corpus[1]. This is a standard dataset in E2E TTS which contains utterances taken from 7 non-fiction books. While it would be interesting to test the effect of morphological features on dialects and languages with more extensive morphological processes than English, S2S TTS models require a comparatively large amount of paired audio and text data, and this was only most conveniently available to us for English.

Out of the total 13,100 utterances in LJ speech (24 hours), we used 9871 utterances with IV items only, totalling approximately 18 hours of speech. We left out utterances containing OOVs to ensure consistent and correct morphological composition was available for each word in every utterance. Although predicting morphological features for OOVs is straightforward and relatively accurate, this would have added unnecessary complication for our purposes here.

We used a Tacotron model implemented by the Github user Fatchord[2]. Tacotron uses a pre-net and CBHG module to encode a series of one-hot input characters from a sequence into a single representation. Unlike previous DNN-based systems, an attention mechanism aligns input text to audio directly, and thus enables grapheme-based input. We used Location Sensitive Attention (LSA) to reduce instability in output speech as recommended in [23].

Each Tacotron was trained for 350k training steps. As [24] noted, neural S2S TTS models occasionally fail to stabilise, as differing random seeds influence the ability to learn effective alignments between text and speech. Some Tacotrons were unstable after training, therefore we trained each system with 3 different seeds and, after informal listening, selected the best performing for the MUSHRA.

This implementation uses a WaveRNN vocoder based on [25]. It was trained using Tacotron's predicted outputs up to 800k steps and synthesised samples in batch-mode. We used a sampling rate of 16kHz.

2.4. MUSHRA Design

To measure the effect of including morphology in neural S2S TTS, we conducted a MUSHRA listening test. We synthesised from 4 systems and used a natural utterance as a hidden reference. We randomly selected 20 utterances from LJ that contained OOVs and added the OOVs with correct morphology to the test text input.

Example input to the 4 systems is shown in Table 1. The baseline is the grapheme-based (G) system, with base graphemes (letters) only. Importantly, the LJ corpus is normalised to expand out numbers and monetary amounts, but acronyms are left unchanged. The graphemes enhanced with morphology (GM) contain the letters interspersed with morphological boundaries.

Table 2: *G2P error (%) with (GM) and without (G) morphemes*

	Random		**Disjoint**	
	WER	PER	WER	PER
G2P_G	9.9	2.3	32.6	5.9
G2P_GM	7.9	1.9	23.4	3.9

In previous work [10], we found phone input significantly improved the quality of Ophelia[3] a different S2S architecture, based on DCTTS [26]. Here, we also measure the effect of morphemes when using phone-based input. In this way, we can measure the improvements of morphology relative to both phones and graphemes. We used Festival as a front-end for the phone-based systems and to enhance input sequences with morphological boundaries.

We held a closely-controlled listening test. We used the BeaqleJS platform [4] to implement a MUSHRA listening test [27] which randomised the systems in a latin-square design. It also ensured every sample was listened to before listeners could proceed. The tests were conducted in purpose-built sound-insulated booths, and playback volume was kept consistent across all tests. We employed 30 native speakers with no known hearing impairments, who were paid £7 to listen to and score 20 utterances from each of the 4 systems over a 45 minute period.

We aggregated the scores across all users and calculated a mean score for each system. We used the student t-test to measure the significance of the aggregated mean of each system, using the Holm-Bonferroni method for error correction. All significance tests were conducted using the *statsmodels* python module.

3. Results

3.1. G2P Error Rate Comparison

The G2P results are presented in Table 2. The WER for the *random* test set appears remarkably low. This is because there is a high amount of root morpheme cross-over between the training and test sets. Around 80% of Unisyn entries are derived from words with pre-existing root morphemes. Yet still the WER was improved by 2%. Moreover, the WER improved by 9.2% on the *disjoint* set containing unseen roots. These results clearly show morphology can improve G2P performance substantially, though not exactly how. Establishing that will be the focus of our next future work.

3.2. MUSHRA Results for TTS Comparison

The average and spread of scores for each system in the MUSHRA is shown in Figure 2. Importantly, the MUSHRA is designed to demonstrate comparative results between systems. As such the score for any one systems should be interpreted relative to another, not as an absolute in itself. The hidden reference for this experiment was natural speech, and all our models exhibit unnatural intonation patterns due to a process of F0 averaging (see [28] and [12]). The overall quality of our speech samples is reasonably high though – we encourage readers to

[1] Available from: https://keithito.com/LJ-Speech-Dataset/

[2] Available from: https://github.com/fatchord/WaveRNN

[3] Available from https://github.com/oliverwatts/ophelia

[4] Available from: https://github.com/ZackHodari/beaqlejs

Table 3: *Improvements in TTS pronunciation by adding morphology: systems G and GM. Listen to speech samples online. The IPA is used to broadly transcribe synthetic speech samples in an American accent.*

G input	GM input	G Pronunciation (Incorrect)	GM Pronunciation (Correct)
coathanger	{coat}{hang}>er>	[kəˈðeɪnʤə]	[ˈkoʊtˌhæŋəɹ]
pothole	{pot}{hole}	[ˈpɑðəl]	[ˈpɑtˌhoʊl]
goatherd	{goat}{herd}	[ˈgɑːðəd]	[ˈgoʊtˌhɜɹd]
loophole	{loop}{hole}	[luːˈfəʊl]	[ˈlupˌhoʊl]
upheld	{up}{held}	[ʌˈfɛld]	[ʌpˈhɛld]
cowherd	{cow}{herd}	[ˈkaʊəɹd]	[ˈkaʊˌhɜɹd]
gigabytes	<giga<{byte}>s>	[gɪˈgaːbɪts]	[ˈgɪgəˌbaɪts]
wobbliest	{wobble}>y>>est>	[ˈwɑblist]	[ˈwɑbliɪst]
optimisers	{optim==ise}>er>>s >	[ˈɑptɪmɪzəz]	[ˈɑptɪmaɪzəɹz]
synchronizable	{syn==chron==ize}>able>	[sɪˈtraɪzəbl]	[sɪŋkrəˈnaɪzəbl]

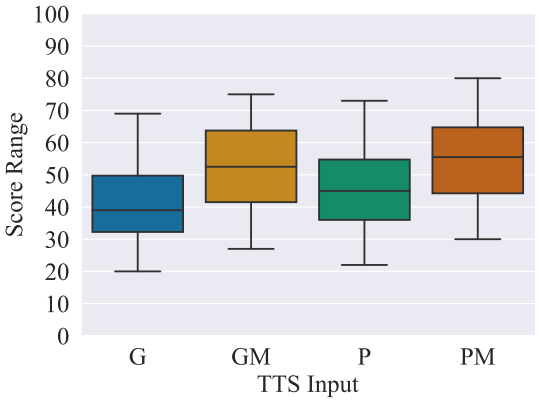

Figure 2: *Range of scores from MUSHRA listening test of each system with phones or grapheme-based input with or without morphology*

listen to samples on our web page[5].

Supplementing both grapheme- and phone-based input with simple boundaries leads to improvements. The differences in mean between systems G and GM are significant with a p-value <0.05. These results show augmenting input with morpheme boundaries also substantially improves neural S2S TTS quality.

Listening to sample output, we find including morphology leads to clearer and more fluent sounding intonation phrases. The system GM also disambiguates pronunciation over certain letter clusters that system G pronounces incorrectly. To demonstrate this, we synthesised words that were typically error-prone for neural S2S G2P and TTS models in [29]. Table 4 shows how adding morphology improves pronunciation of such words as *coathanger*, *upheld*, and *wobbliest*.

System P system outperformed system G, repeating our previous findings from [10] with a different Tacotron implementation and vocoder. This is because phones map pronunciations directly unlike letters, leading to fewer ambiguities at training and test time. In this respect, it is also remarkable that system GM outperformed system P, which suggests knowledge of word formation from morphology contributes more to TTS quality than using phones.

[5]Listen at: http://homepages.inf.ed.ac.uk/s1649890/morph/

Table 4: *Improvements in TTS pronunciation from using phones: systems GM and PM*

Word	GM (Incorrect)	PM (Correct)
untypable	[ˈʌntɪpəbl]	[ʌnˈtaɪpəbl]
pyjama	[ˈpæʤəmə]	[pəˈʤæmə]
flaubert	[ˈflɑːbət]	[fləʊˈber]
karate	[kəˈreɪt]	[kəˈrɑːti]
eduardo	[eˈdɔrdu]	[eˈdwɑːrdəʊ]
macao	[meɪˈkeʊ]	[məˈkaʊ]
crimea	[ˈkraɪmi]	[kraɪˈmiːə]
labyrinth	[ˈleɪbəɹɪnθ]	[ˈlæbərɪnθ]
ASCII	[əˈsiː]	[ˈæski]

3.3. How useful are phones?

Our results indicate system GM performed almost equally to system PM. The MUSHRA evaluates systems generally, and improvements in phrasing due to morphological boundaries lead to similar sounding systems. This suggests simple morphological boundaries could replace the expensive-to-maintain pipeline of generating accurate phone sequences for a training transcript. Moreover in [16], a Bi-LSTM with grapheme input predicted morphological boundaries with an accuracy of 91.1% for Uniysn and 93.8% for Combilex. Similar results were found in other languages in [30]. We intend to evaluate the difference between using predicted and oracle morphological boundaries as input to neural S2S TTS systems in future work.

However, we note it is still the case at test time or in deployment that specific cases of letter-to-sound (LTS) ambiguities might require disambiguation via phones. This is particularly true with rare LTS relations, as in foreign names. Table 4 shows some examples where the GM system produced an incorrect pronunciation.

4. Conclusions

We measured effects of morphological features on S2S models used in TTS. By improving the productivity of training data and resolving ambiguous letter-to-sound relations we achieved substantial gains in G2P and TTS. We found morphological features significantly increased naturalness scores of grapheme-based systems to the level of phone-based systems. When access to phonetic transcriptions is limited, morphological labels still provide much of the benefit on sum. But they do not correct all specific pronunciation errors.

5. References

[1] CMU, "The Carnegie Mellon pronouncing dictionary," 2020. [Online]. Available: https://github.com/cmusphinx/cmudict

[2] S. Fitt, "Unisyn lexicon," 2020. [Online]. Available: http://www.cstr.ed.ac.uk/projects/unisyn/

[3] S. Fitt and K. Richmond, "Redundancy and productivity in the speech technology lexicon - can we do better?" in *Proc. Interspeech*. ISCA, 2006, pp. 1202–1204.

[4] R. Clark, K. Richmond, and S. King, "Multisyn: Open-domain unit selection for the Festival speech synthesis system," *Speech Communication*, vol. 49, no. 4, pp. 317–330, 2007.

[5] M. Schröder and J. Trouvain, "The German text-to-speech synthesis system MARY: A tool for research, development and teaching," *International Journal of Speech Technology*, vol. 6, no. 4, pp. 365–377, 2003.

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

[7] Y. Wang *et al.*, "Tacotron: Towards end-to-end speech synthesis," in *Proc. Interspeech*, 2017, pp. 4006–4010.

[8] O. Watts *et al.*, "Where do the improvements come from in sequence-to-sequence neural TTS?" in *Proc. 10th ISCA Speech Synthesis Workshop*, 2019, pp. 217–222.

[9] T. Hayashi *et al.*, "ESPnet-TTS: Unified, reproducible, and integratable open source end-to-end text-to-speech toolkit," in *ICASSP*, 2020, pp. 7654–7658.

[10] J. Fong *et al.*, "A comparison of letters and phones as input to sequence-to-sequence models for speech synthesis," in *Proc. 10th ISCA Speech Synthesis Workshop*, 2019, pp. 223–227.

[11] H. Guo *et al.*, "Exploiting syntactic features in a parsed tree to improve end-to-end TTS," in *Proc. Interspeech*, 2019, pp. 4460–4464.

[12] T. Kenter *et al.*, "CHiVE: Varying prosody in speech synthesis with a linguistically driven dynamic hierarchical conditional variational network," in *Proc. ICML*, vol. 97, 2019, pp. 3331–3340.

[13] W. Fang, Y.-A. Chung, and J. Glass, "Towards transfer learning for end-to-end speech synthesis from deep pre-trained language models," 2019. [Online]. Available: http://people.csail.mit.edu/andyyuan/docs/tacobert.paper.pdf

[14] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proc. ACL*, 2016, pp. 1715–1725.

[15] K. Ito, "The LJ speech dataset." [Online]. Available: https://keithito.com/LJ-Speech-Dataset/

[16] M. Bikmetova and K. Richmond, "Neural network modelling of grapheme-to-phoneme transformations: automatic morphological analysis," *Bulletin of UGATU*, vol. 84, no. 2, pp. 121–126, 2019.

[17] K. Kann, R. Cotterell, and H. Schütze, "Neural morphological analysis: Encoding-decoding canonical segments," in *Proc. EMNLP*, 2016, pp. 961–967.

[18] S. Fitt, "Morphological approaches for an English pronunciation lexicon," in *Proc. Eurospeech*, 2001, pp. 1069–1072.

[19] H. Sun *et al.*, "Token-level ensemble distillation for grapheme-to-phoneme conversion," in *Proc. Interspeech*, 2019, pp. 2115–2119.

[20] K. Rao *et al.*, "Grapheme-to-phoneme conversion using long short-term memory recurrent neural networks," in *Proc. ICASSP*, 2015, pp. 4225–4229.

[21] G. Klein *et al.*, "OpenNMT: Open-source toolkit for neural machine translation," in *Proc. ACL*, 2017, p. 67–72.

[22] M. Luong, H. Pham, and C. Manning, "Effective approaches to attention-based neural machine translation," in *Proc. ACL*, 2015, pp. 1412–1421.

[23] J. Shen *et al.*, "Natural TTS synthesis by conditioning WaveNet on Mel spectrogram predictions," in *Proc. ICASSP*, 2018, pp. 4779–4783.

[24] J. Latorre *et al.*, "Effect of data reduction on sequence-to-sequence neural TTS," in *Proc. ICASSP*, 2019, pp. 7075–7079.

[25] N. Kalchbrenner *et al.*, "Efficient neural audio synthesis," in *Proc. ICML*, vol. 80, 2018, pp. 2410–2419.

[26] H. Tachibana, K. Uenoyama, and S. Aihara, "Efficiently trainable Text-to-Speech system based on deep convolutional networks with guided attention," in *Proc. ICASSP*, 2018, pp. 4784–4788.

[27] U. Z. S. Kraft, "Beaqlejs: HTML5 and javascript based framework for the subjective evaluation of audio quality," in *Linux Audio Conference*, 2014, pp. 2095–2099.

[28] O. W. Zac Hodari and S. King, "Using generative modelling to produce varied intonation for speech synthesis," in *To appear in Proc. Speaker Odyssey*, 2020. [Online]. Available: https://arxiv.org/abs/1906.04233

[29] J. Taylor and K. Richmond, "Analysis of Pronunciation Learning in End-to-End Speech Synthesis," in *Proc. Interspeech*, 2019, pp. 2070–2074.

[30] D. Sharma, M. Wilson, and A. Bruguier, "Better Morphology Prediction for Better Speech Systems," in *Proc. Interspeech*, 2019, pp. 3535–3539.