10th ISCA Speech Synthesis Workshop
20-22 September 2019, Vienna, Austria

A Comparison between Letters and Phones as Input to Sequence-to-Sequence Models for Speech Synthesis

Jason Fong, Jason Taylor*, Korin Richmond and Simon King*

The Centre for Speech Technology Research (CSTR), University of Edinburgh, UK

{jason.fong, jason.taylor, korin.richmond, Simon.King}@ed.ac.uk

Abstract

Neural sequence-to-sequence (S2S) models for text-to-speech synthesis (TTS) may take letter or phone input sequences. Since for many languages phones have a more direct relationship to the acoustic signal, they lead to improved quality. But generating phone transcriptions from text requires an expensive dictionary and an error-prone grapheme-to-phoneme (G2P) model, and the relative improvement over using letters has yet to be quantified. In approaching this question, we presume that letter-input S2S models must implicitly learn an internal counterpart to G2P conversion and therefore inevitably make errors. Such a model may thus be viewed as phone-input S2S with inaccurate phone input. To quantify this inaccuracy, we compare in this paper a letter-input S2S system to several phone-input systems trained on data with a varying level of error in the phonetic transcription. Our findings show our letter-input system is equivalent in quality to the phone-input system in which 25% of word tokens in the training data have incorrect phonetic transcriptions. Furthermore, we find that for phone-input systems up to 15% of word tokens in the training data can have incorrect phonetic transcriptions without any significant difference in performance to a 0% error rate system. This suggests it is acceptable to use G2P to predict pronunciations for out-of-vocabulary words (OOVs) provided they are less than around 15% of the training data, removing the need to manually add OOVs to the dictionary for every new training set.

Index Terms: Speech Synthesis, Sequence-to-Sequence, End-to-End, Grapheme-to-Phoneme

1. Introduction

A fundamental task in text-to-speech synthesis (TTS) is disambiguating complex grapheme-to-acoustic (G2A) relations. For historical reasons, English graphemes do not intuitively represent the pronunciation, and consequently the sound, of words. For instance, *'sing'* and *'sine'* only share the same sound of their first grapheme *'s'* despite only differing by their final graphemes *'g'* and *'e'*. The standard approach to dealing with this in statistical parametric speech synthesis (SPSS) systems for many years has been to employ forced-alignment to sequentially delimit and align segments of speech using linguistic symbols in order to train an acoustic model. The more monotonic and predictable the mapping is between linguistic symbols and speech segments in the training data, the higher the quality of the resulting model. Unlike graphemes, phonetic transcriptions deliberately map to the sounds of a language in a consistent and monotonic fashion. Thus, phones approximate speech more closely than graphemes. Generating phone sequences corresponding to input text is then typically handled by a sequence of processing modules chained together, for example in front-end packages such as Festival [1], Mary [2] or Sparrowhawk [3]. The front-end typically includes a pronunciation lexicon lookup and a grapheme-to-phoneme (G2P) model for dealing with out-of-vocabulary words (OOVs), phonetic disambiguation of non-standard words such as numbers, abbreviations and homographs, and post-lexical rules such as the addition of intrusive-r in Received Pronunciation [4]. Development of these modules requires a large amount of manual expertise and effort. This is especially true of building a lexicon. This cost means that TTS technology is commercially viable only for the world's most widely spoken languages. Although there are over 7000 languages spoken around the world [5], high quality commercial voices exist for just over 30 of them [6, 7].

In a bid to avoid such high effort and expense, recent work on neural sequence-to-sequence (S2S) models proposes training with grapheme inputs directly [8], avoiding the need for explicit front-end modules. Unlike systems dependent on forced-alignment, S2S models employ a text encoder with an attention mechanism that learns pronunciation information by considering an entire input sequence in a non-monotonic fashion. This means that the acoustics are dynamically modelled on a longer contextual input window than encompassed by incremental HTS-style labels [9] in feedforward DNN-based systems such as Merlin [10]. This distinction is depicted in Figure 1. The pronunciation of grapheme clusters like 'gh' in *tou**gh*** and *thou**gh*** is more likely to be successfully predicted from graphemes in a S2S model, as the context of the entire word can be used rather than graphemes consumed incrementally.

However, evidence from multiple studies (e.g. [11, 12, 13]) suggests a monolithic S2S model learns a weaker joint pronunciation and acoustic model from graphemes than phones. Several reasons could explain this. First, graphemes in the training data can denote speech sounds ambiguously. For instance, homographs and numbers may have separate pronunciations depending on the surrounding semantic or syntactic context. Additionally, grapheme clusters may be pronounced differently depending on the surrounding graphemic context, for example the bold letters in *tou**gh***, *w**o**men* and *na**ti**on* represent different sounds from the same graphemes in *thou**gh***, *w**o**mb* and *na**ti**ve*. It is clear graphemes, unlike phones, introduce a high level of uncertainty in expected pronunciations. Second, TTS training corpora usually contain fewer word types than a lexicon. This means the implicit pronunciation model in a letter-input S2S system may be exposed to fewer foreign names and loan words and thus less irregular G2A relations such as in words like *Flaubert* or *baguette*. Furthermore, words (and their G2A relations) in training corpora for S2S models are distributed with unbalanced frequencies when compared to lexica used for training G2P models (in which each word type occurs exactly once). Rare words, or grapheme sequences with unusual G2A relations, occur far less frequently in S2S training corpora than common vocabulary and functional words, such as articles

*These authors contributed equally to this work.

10.21437/SSW.2019-40

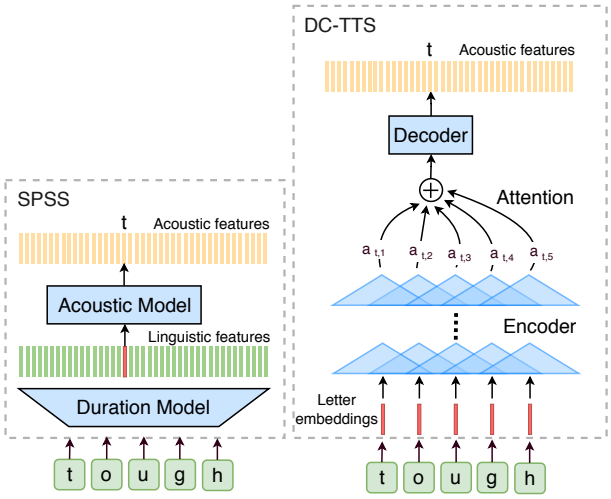

Figure 1: *Comparison between the context window size that is available to the Acoustic Model / Decoder in SPSS and DC-TTS. Highlighted in red are the input features available when predicting acoustic features at time 't'. The decoder in DC-TTS has access to all the graphemes in the input sequence due to the convolutional layers in the encoder and attention mechanism. Conversely the SPSS acoustic model takes only one frame of linguistic features as input, exactly how many letters this spans depends on the nature of the contextual letter features it contains.*

(*a*, *the*) and prepositions (*for*, *on*, etc.).

Each of these reasons is likely to contribute to the degradation observed in grapheme-based systems, and could vary depending on the size and style of the dataset used. When building an English S2S TTS system we face a dilemma between investing in front-end resources to improve pronunciation or using graphemes as input and accepting less than optimal voice quality and pronunciation accuracy.

There has been little prior work on directly measuring the gap in performance between different input forms for S2S TTS models. [11] compared systems where the input representation was mixed between graphemes and phones simultaneously. Although they did not train systems separately with either graphemes or phonemes alone, they noted that phonetic information improved overall voice quality. The Tacotron2 team [12] noted that acoustic quality was poorer when using graphemes, with mispronunciations and incorrect prosody found to be among the system's most common errors. This was shown again in [13], where words with ambiguous relationships between their orthographic form and phonetic realisation (e.g. foreign names and loan words) were especially problematic for another S2S model.

We aim to build on prior findings here by quantifying overall degradation when training with graphemes in place of phonetic input. In addition, to aid our understanding of system performance with graphemes, it is compared to phone-based equivalents with varying proportions of phonetic corruption. By truncating a lexicon and phonetising increasingly larger proportions of the training data with G2P modules of differing quality, we obtain systems that simulate the G2A ambiguity that occurs when using graphemes. We then rank the systems together by measuring their naturalness in a MUSHRA evaluation.

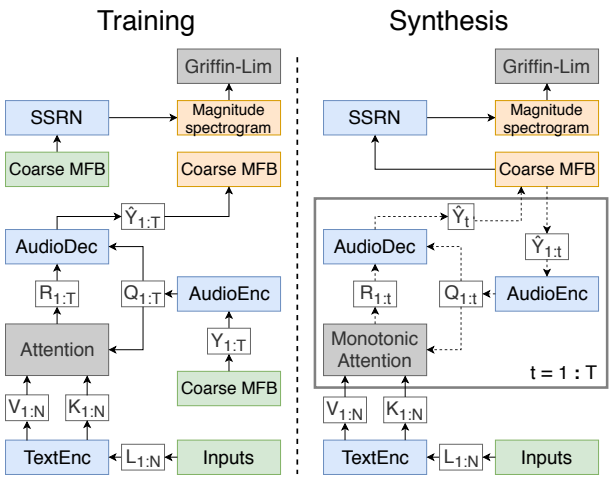

Figure 2: *Overview of the DC-TTS architecture used in our system. Blue: learned modules. Grey: operations. Green: inputs. Orange: predictions. Plate notation over $t = 1 : T$ denotes the autoregressive loop at synthesis time.*

2. Experiments

2.1. TTS Model Architecture

We trained deep convolutional neural sequence-to-sequence TTS models [14] with both phone and grapheme inputs using Ophelia [15]. This architecture performs two sequential tasks to produce speech spectrograms from linguistic inputs, as shown in Figure 2. First, the Text-to-Mel network (T2M) uses sequence-to-sequence text and audio encoders with attention architecture [16] to consume the inputs and predict 'coarse-in-time' mel-spectrograms. The Super Spectrogram Resolution Network (SSRN) then consumes these mel-spectrograms and upsamples them in both time and frequency to produce a full magnitude spectrogram. Finally, the Griffin-Lim algorithm is used to re-introduce phase to the magnitude spectrogram and thus create the output speech waveform.

All systems use distinct T2M modules trained for 500 epochs on their respective input training data and share the same SSRN module, also trained for 500 epochs. Forcibly incremental attention, labelled as 'Monotonic Attention' in Figure 2, is used at test time but not training time. At test time each trained system uses the same 0% word error rate (WER) test transcript, rather than test transcripts corrupted in the same way as their training transcripts. In the context of G2P, a word error occurs when at least one phone in the word is incorrectly predicted. Further details regarding the system architecture and hyperparameter setup can be found in [17].

2.2. Speech Data

We used paired text-audio data from the Linda Johnson (LJ) corpus [18] which contains 24 hours of audio distributed over 13,100 utterances taken from 7 non-fiction books. The text has been normalised to expand out numbers, ordinals, and monetary amounts, but acronyms are left unchanged. We trained S2S TTS models using a subset of 9871 utterances from this corpus. Utterances which contained OOV's according to Combilex were removed to allow for a training transcript where all words were in-vocabulary (IV). 242 test utterances (IV ones from chapter 50) were held out from training for use in our listening test.

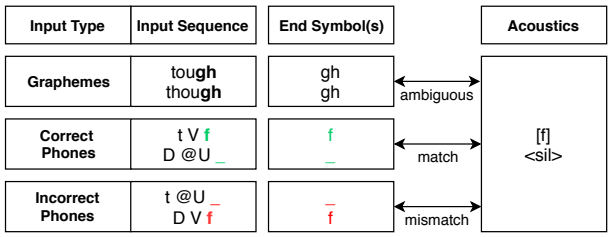

Figure 3: *Nature of mapping from various input letter sequences to acoustics. The underscore symbol '_' represents a missing phone from the prediction, and '<sil>' represents a missing sound from the acoustics. Using correctly predicted phones results in a match between inputs and acoustics during training that should produce a high quality acoustic model at test time. However using incorrectly predicted phones results in a mismatch that may negatively impact the performance of the acoustic model. As the mapping between graphemes and acoustics can be rather ambiguous given the semantic, syntactic, or graphemic context, we would also expect the use of graphemes to have a negative effect on the acoustic model.*

2.3. Lexicon and G2P models

The General American (GAM) surface-form of the Combilex speech technology lexicon [19] was used to phonetise the LJ transcript, and to train a classification and regression tree (CART) as well as neural G2P models. The GAM surface-form is a version of Combilex tailored to LJ's speech. Combilex was selected over the widely used Carnegie Mellon Pronouncing Dictionary [20], for its higher consistency and accuracy. The lexicon and G2P models were integrated into Festival's standard front-end pipeline. The G2P models were used when the GAM surface-form lexicon was truncated to deliberately induce phone corruption, as explained in Section 2.4.

The neural G2P model was a Bidirectional Long Short Term Memory (BLSTM) network from [13] built using OpenNMT and Pytorch [21]. We used 6 bi-directional encoder and decoder layers with 500 units each, a learning rate of 0.0001, and Luong's global attention [22] with dropout of 0.1. The network was trained with mini-batches of 64 and optimised with ADAM. The BLSTM converged after 50,000 training steps.

The CART model was built using Festival, based on the system described in [23]. This model was trained with phone-grapheme alignment information provided in Combilex.

2.4. Creation of Training Transcripts

Figure 3 shows how grapheme input is ambiguous in its relations to acoustics and how the ambiguity may be approximated by using incorrect phones. We see that the letter cluster 'gh' may represent either silence or a voiceless labio-dental fricative. The sounds corresponding to these letters may potentially be reversed with errors from a G2P model. Therefore, we generate a range of **phonetic** transcripts with increasing proportions of incorrect phones to serve as a proxy for assessing the errors likely to arise when using **graphemic** input.

We create phone transcripts using the following methodology. First, we generate a 'gold standard' 0% WER transcript via look-up using the full lexicon. This is as close as we can get to a linguist's transcription given the large dataset sizes used in S2S TTS. Second, we generate inferior phonetic transcriptions by varying the proportion of the training text phonetised by lexicon or by differing G2P models. In this way we compare the grapheme-based system to phone-based systems with varying consistency in the phones-to-acoustics relations.

Table 1: *System Description.* ***Input*** *column denotes method used to phonetise transcript.* ***Ratio*** *column denotes % of full lexicon used, or the entries replaced by G2P predictions. The* ***WER*** *is calculated as the percentage of words containing a phone error in the transcript.*

Name	Input	Ratio (%)	WER (%)
`100combi`	Lexicon Lookup (LL)	100	0.0
`50neur`	LL / Neural G2P	50 / 50	11.5
`50cart`	LL / CART G2P	50 / 50	14.3
`100neur`	Neural G2P	100	25.2
`100cart`	CART G2P	100	30.6
`let`	Graphemes	100	-

Table 1 shows the breakdown of input data used to train each system. The '**Ratio**' column details the proportion of each input type to each system. The '**WER**' column shows the Word Error Rate of each training transcript. The error rates reported for the G2P models were much lower in their respective papers (the CART and neural model's scoring $<$15% WER), this is because those error rates are calculated over their **test** sets (which are held out entries from their pronunciation lexica), which do not reflect the natural frequencies of word types in our **training** LJ speech transcript. This difference is an important consideration when integrating a G2P model into a developed TTS system, as the WER of an abstract test set may not reflect the WER on natural occurrences of words as employed in the TTS domain.

Importantly, some G2P errors may in fact be plausible variants, such as a prediction of the word 'tamil' with [I] (in Combilex's symbolic representation) instead of a stated schwa phone [@] in the lexicon. This is not a gross error that would degrade TTS quality noticeably. The extent of this effect is however unquantified and has not been measured as it would require a manual review of all errors in the transcripts.

2.5. Listening Test

The six systems in Table 1 were submitted to a MUSHRA test with natural recordings (copy synthesis) used as a sanity check to ensure a correct upperbound. We recruited 30 English native speakers as listeners, paid £8 each for 45 minutes. The listening tests were conducted in purpose-built listening booths.

The focus of our inquiry was the training transcript and its effect on the quality of a built voice. Hence, only words with mappable pronunciations (i.e. easily predictable from graphemes) were used at test time. We hand-selected 20 utterances of such words from our 242 test utterances. These utterances did not require any disambiguation via the traditional front-end as homographs and abbreviations were excluded and numbers verbalised. In this way, any errors resulting from grapheme ambiguity in test utterance words were minimised. For a fair comparison of the models learnt from the training transcripts alone, we used the same Combilex transcriptions from the complete lexicon for testing all phone-based systems.

3. Results

3.1. System Comparison

Figure 4 displays the results of the MUSHRA test. Participants were instructed to raise the score of the highest quality voice (natural) to 100, which is clearly evident in the results. No such

stipulations were made for other voices, though, and all systems score below 52% on the naturalness scale, including that trained with a transcript with 0% WER. Whilst we evaluated the general performance of each model by varying the input to the T2M network, artefacts resulting from the use of the Griffin-Lim algorithm are likely to have influenced the average score for each system. It is also possible that finer differences between the voices could have been masked by the Griffim-Lim artefacts. This scale only measures naturalness, but we hypothesise that intelligibility may be affected by Griffim-Lim distortion. If a higher quality replacement of Griffim-Lim were used to re-introduce phase, analysing whether this hypothesis is true would be unnecessary. We therefore aim to replace Griffim-Lim for future analysis of S2S TTS models.

There is a 23.5% relative drop in the naturalness score in the S2S model when trained on graphemes (`let`) rather than phones from the full lexicon (`100combi`), from 51.8 to 39.6. They are significantly different with $p < 0.0005$. In terms of phonetic corruption meanwhile, the performance `let` is non-significantly different to a phone-based system trained with 25% WER (`100neur`) which has a naturalness rating of 39.1.

The differences between the three best performing systems `100combi`, `50cart`, and `50neur` are negligible and not significant. While this equivalence could be interpreted as suggesting that training transcripts with a WER corruption of up to 15% bear negligible degradation generally, in reality a large proportion of the words in fact have viable pronunciations. That is, incorrectly predicted phones could still be acoustically similar to the corresponding speech data and thus will not greatly degrade the acoustic model. Unfortunately, teasing apart which predictions are viable and which are implausible (i.e. would have a larger detrimental effect in training) is non-trivial. Furthermore, the differences between these systems may be caused by other factors such as Griffim-Lim or the particular random seed that initialises the parameters of DC-TTS during training.

The grapheme based system `let`, as well as `100cart` and `100neur` synthesise speech less "crisply". Furthermore, there are noticeable pronunciation errors in synthesised test utterances unlike in the other systems (even though correct phones are being used at test time). Examples of pronunciation errors may be heard on our samples [1]. This demonstrates that ambiguous and indirect input-to-acoustics mappings at training time leads to demonstrable degradation on acoustic quality at test time.

3.2. OOVs in Phone-based Systems

In training we excluded utterances with Combilex OOVs. This was to ensure the lexicon contained pronunciations for all word tokens present to build `100combi`. While it is inherently difficult to measure the effect of using G2P to predict pronunciations for error-prone OOVs, the relatively strong performance of `50neur` and `50cart` suggest that OOVs in large datasets could generally be phonetised by a G2P model without a severe impact on acoustic quality.

As Table 2 shows, although approximately a quarter of utterances in the LJ dataset contain an OOV, <2% of total tokens are OOV. Even though `50neur` and `50cart` possess a WER of up to 14.3% across total tokens, they still rendered similar performance to the system trained with all tokens being IV (`100combi`). Even if all OOV tokens were mis-phonetised in LJ, is is plausible to hypothesise that the overall effect on S2S TTS performance would be negligible.

[1] https://jonojace.github.io/SSW19-comparison

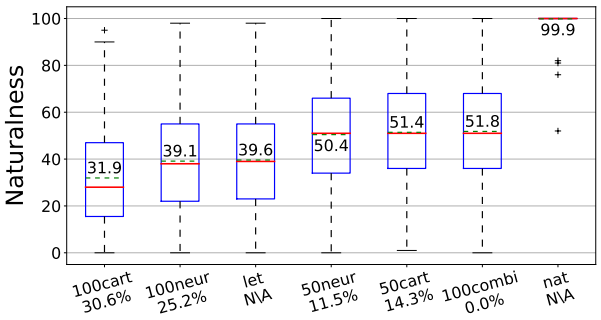

Figure 4: *MUSHRA results. Solid red lines are **medians**, dashed green lines are **means** (also numerically labeled), blue boxes show the **25th** and **75th percentiles**, and whiskers show the **range** of the ratings, excluding **outliers** which are plotted with +. Percentages below system names indicate phone WER of their respective training transcript.*

Table 2 also shows how OOVs are distributed in other corpora used for training TTS models. Even in a more phonetically diverse corpus such as Nancy (from the Blizzard Challenge 2011 [24]), or a large corpus of audiobooks like LibriTTS [25], OOV token rates of 4.9% and 1.3% respectively suggest that using a G2P model for these OOV tokens during training would not be detrimental to a phone-based system. In addition, the full dataset may then be utilised in training, compared with training with only IV utterances. The hypothesis of whether labelling OOVs in non-detrimental and/or effective is the object of future study.

Table 2: *Combilex OOVs in large TTS datasets. The **type** and **token** rates describe how many individual word types and tokens are OOV. The **utt** rate is the percentage of utterances containing at least one OOV token.*

Dataset	Hours	OOV Type Rate (%)	OOV Token Rate (%)	OOV Utt Rate (%)
LJ	24	9.8	1.9	24.6
Nancy	17	10.5	4.9	56.9
LibriTTS	585	32.8	1.3	18.0

4. Conclusions

Graphemes in English have a more ambiguous relationship with acoustics than phones. Text encoders and decoders with attention mechanisms in S2S TTS models lend themselves to coping with the non-monotonic and unpredictable nature of G2A relations in English, however training with grapheme input still does not work as well as phone input.

We sought to quantify the gap in performance between grapheme-based and phone-based S2S TTS models by performing a MUSHRA evaluation with phone-based systems trained with differing amounts of incorrect phones as generated by standard G2P models.

The grapheme-based system performed significantly worse than a system with 0% WER training transcript, and roughly equivalent to a system with a 25% WER. However, a <15% WER training transcript performed with negligible differences to a 0% WER one, suggesting that OOV tokens in training transcripts may be phonetised by a G2P model. We will explore this suggestion in future work.

5. References

[1] R. A. J. 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.

[2] M. Schrder 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.

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

[4] P. Taylor, *Text-to-Speech Synthesis*. Cambridge: Cambridge University Press, 2009.

[5] Ethnologue, "How many languages are there in the world?" 2019. [Online]. Available: https://www.ethnologue.com/guides/how-many-languages

[6] Amazon, "Voices in Amazon Polly," 2019. [Online]. Available: https://docs.aws.amazon.com/polly/latest/dg/voicelist.html

[7] Google, "Cloud Text-to-Speech," 2019. [Online]. Available: https://cloud.google.com/text-to-speech/

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

[9] H. Zen, "An example of context-dependent label format for HMM-based speech synthesis in English," 2006. [Online]. Available: https://wiki.inf.ed.ac.uk/twiki/pub/CSTRF0parametrisation/hts_lab_format.pdf

[10] Z. Wu, O. Watts, and S. King, "Merlin: An open source neural network speech synthesis system," in *9th ISCA Speech Synthesis Workshop, Proceedings*, 2016.

[11] K. Kastner *et al.*, "Representation mixing for TTS synthesis," in *ICASSP - IEEE International Conference on Acoustics, Speech and Signal Processing, Proceedings*, 2019, pp. 5906–5910.

[12] J. Shen *et al.*, "Natural TTS synthesis by conditioning WaveNet on Mel spectrogram predictions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Proceedings*, 2018, pp. 4779–4783.

[13] J. Taylor and K. Richmond, "Analysis of pronunciation learning in end-to-end speech synthesis," in *To appear in Interspeech 2019*, 2019. [Online]. Available: http://homepages.inf.ed.ac.uk/s1649890/lts/

[14] H. Tachibana, K. Uenoyama, and S. Aihara, "Efficiently trainable Text-to-Speech system based on deep convolutional networks with guided attention," in *ICASSP - IEEE International Conference on Acoustics, Speech and Signal Processing, Proceedings*, 2018, pp. 4784–4788.

[15] O. Watts, "Ophelia: A modified version of Kyubyong Park's DC-TTS repository, which implements a variant of the system described in Efficiently Trainable Text-to-Speech System Based on Deep Convolutional Networks with Guided Attention," May 2019. [Online]. Available: https://github.com/oliverwatts/ophelia

[16] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *ICLR - International Conference on Learning Representations, Proceedings*, 2015.

[17] J. Fong *et al.*, "Investigating the robustness of sequence-to-sequence text-to-speech models to imperfectly-transcribed training data," in *To appear in Interspeech 2019*, 2019. [Online]. Available: https://jonojace.github.io/papers/IS19-robustness.pdf

[18] K. Ito, "The LJ speech dataset," 2017. [Online]. Available: https://keithito.com/LJ-Speech-Dataset/

[19] K. Richmond, "Combilex speech technology lexicon," 2018. [Online]. Available: http://homepages.inf.ed.ac.uk/korin/sitenew/Research/Combilex

[20] CMU, "The Carnegie Mellon pronouncing dictionary," 2019. [Online]. Available: https://github.com/cmusphinx/cmudict

[21] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush, "OpenNMT: Open-source toolkit for neural machine translation," in *ACL 2017 - Annual Meeting of the Association for Computational Linguistics, Proceedings*, 2017, pp. 67–72.

[22] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *ACL - Annual Meeting of the Association for Computational Linguistics, Proceedings*, 2015, pp. 1412–1421.

[23] K. Richmond, R. A. J. Clark, and S. Fitt, "Robust LTS rules with the Combilex speech technology lexicon," in *Interspeech, Proceedings*, 2009, pp. 1295–1298.

[24] CSTR, "The Nancy corpus," 2011. [Online]. Available: http://www.cstr.ed.ac.uk/projects/blizzard/2011/lessac_blizzard2011/

[25] H. Zen *et al.*, "LibriTTS: A corpus derived from LibriSpeech for Text-to-Speech," in *Submission to Interspeech*, 2019. [Online]. Available: https://arxiv.org/pdf/1904.02882.pdf