Pronunciation Modelling in End-to-End Text-to-Speech Synthesis

Jason Taylor

Doctor of Philosophy

University of Edinburgh

2022

Abstract

Sequence-to-sequence (S2S) models in text-to-speech synthesis (TTS) can achieve high-quality naturalness scores without extensive processing of text-input. Since S2S models have been proposed in multiple aspects of the TTS pipeline, the field has focused on embedding the pipeline toward End-to-End (E2E-) TTS where a waveform is predicted directly from a sequence of text or phone characters. Early work on E2E-TTS in English, such as Char2Wav [1] and Tacotron [2], suggested that phonetisation (lexicon-lookup and/or G2P modelling) could be implicitly learnt in a text-encoder during training. The benefits of a learned text encoding include improved modelling of phonetic context, which make contextual linguistic features traditionally used in TTS pipelines redundant [3]. Subsequent work on E2E-TTS has since shown similar naturalness scores with text- or phone-input (e.g. as in [4]). Successful modelling of phonetic context has led some to question the benefit of using phone- instead of text-input altogether (see [5]).

The use of text-input brings into question the value of the pronunciation lexicon in E2E-TTS. Without phone-input, a S2S encoder learns an implicit grapheme-to-phoneme (G2P) model from text-audio pairs during training. With common datasets for E2E-TTS in English, I simulated implicit G2P models, finding increased error rates compared to a traditional, lexicon-based G2P model. Ultimately, successful G2P generalisation is difficult for some words (e.g. foreign words and proper names) since the knowledge to disambiguate their pronunciations may not be provided by the local grapheme context and may require knowledge beyond that contained in sentence-level text-audio sequences. When test stimuli were selected according to G2P difficulty, increased mispronunciations in E2E-TTS with text-input were observed. Following the proposed benefits of subword decomposition in S2S modelling in other language tasks (e.g. neural machine translation), the effects of morphological decomposition were investigated on pronunciation modelling. Learning of the French post-lexical phenomenon *liaison* was also evaluated.

With the goal of an inexpensive, large-scale evaluation of pronunciation modelling, the reliability of automatic speech recognition (ASR) to measure TTS intelligibility was investigated. A re-evaluation of 6 years of results from the Blizzard Challenge was conducted. ASR reliably found similar significant differences between systems as paid listeners in controlled conditions in English. An analysis of transcriptions for words exhibiting difficult-to-predict G2P relations was also conducted. The E2E-ASR Transformer model used was found to be unreliable in its transcription of difficult G2P

relations due to homophonic transcription and incorrect transcription of words with difficult G2P relations. A further evaluation of representation mixing in Tacotron finds pronunciation correction is possible when mixing text- and phone-inputs. The thesis concludes that there is still a place for the pronunciation lexicon in E2E-TTS as a pronunciation guide since it can provide assurances that G2P generalisation cannot.

* * *

[Before the dawn of 2001] there will be no C, X or Q in our every-day alphabet. They will be abandoned because unnecessary.

JOHN ELFRETH WATKINS, 1900

Acknowledgements

I would like to thank Korin Richmond for his guidance and support. He always encouraged me to do my best and has truly been all I could have wished for in a supervisor. I would also like to thank my secondary supervisor Steve Renals whose advice guided me out of a particularly difficult time in my first year.

The Centre for Speech Technology Research (CSTR) has provided an excellent environment to source and support novel research ideas. In particular, I would like to thank its director Simon King from whom I have learnt a great deal. In particular I am grateful for his guidance when I tutored the Speech Processing and Speech Synthesis courses. I would also like to thank the other members of CSTR with whom I met frequently during the Speak and Listen discussion groups and during the weekly CSTR talks. In particular I want to extend thanks to those members with whom I have collaborated during the past 4 years including Jason Fong, Oliver Watts and Cassia Valentini-Botinhao. I would also like to thank other researchers in the field of TTS with whom I have had useful discussions - in particular Sébastien Le Maguer and Gustav Eje Henter.

Observing the hard-work and devotion of students I tutored provided inspiration to press-on during difficult periods during the PhD at no time more so than during the COVID-19 pandemic. Their determination to keep going and complete their studies in spite of physical isolation and continuous online learning rubbed off, fuelling me in the past 16 months to complete this thesis. I would also like to give special thanks in particular to 3 excellent MSc students for whom I was their supervisor while they wrote their dissertations: Daniel Jordan, Caela Northey and Annika Viehoff.

I would like to thank all of the passing members of Office 4.23 during my time in the Informatics Forum who helped lightened the atmosphere, in particular those who would greet me every morning and annoyingly switch on the lights as I was slogging away in the dark: Ratish Pupudully, Chao Peng, Vanya Yaneva and Sefa Akca.

I would like to thank the Economic and Social Research Council (ESRC) and the Scottish Graduate School for Social Science (SGSSS) for the financial and academic support I have received. In particular I thank Mhairi Mackenzie of the SGSSS for her advice and suggestions for my research pathways.

Finally, I could not have studied for this thesis without my family who have been behind me every step of the way.

* * *

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Jason Taylor*)

Table of Contents

* * *

List of Figures

* * *

List of Tables

* * *

Publication List

Analyses and experiments contained in this thesis appear in the following works:

Chapter 2

The theoretical analysis of implicit pronunciation modelling appeared in:

J. Taylor and K. Richmond. "Analysis of Pronunciation Learning in End-to-End Speech Synthesis" in *Proc. Interspeech*, 2019, pp. 2070–2074. Available: `http://dx.doi.org/10.21437/Interspeech.2019-2830`

The MUSHRA listening test results comparing text- and phone-inputs appeared in:

J. Fong, **J. Taylor**, K. Richmond, and S. King. "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. Available: `http://dx.doi.org/10.21437/SSW.2019-40`

Chapter 3

The analysis on morphology was presented appeared in :

J. Taylor and K. Richmond. "Enhancing Sequence-to-Sequence Text-to-Speech Using Morphology", in *Proc. Interspeech*, 2020, pp. 1738-1742. Available: `http://dx.doi.org/10.21437/Interspeech.2020-1547`

The analysis on Tacotron in French will appear at SSW 2021:

J. Taylor, S. Le Maguer, and K. Richmond. "Liaison and Pronunciation Learning in End-to-End Text-to-Speech in French", To appear *11th ISCA Speech Synthesis Workshop*, 2021. Available: `http://homepages.inf.ed.ac.uk/s1649890/fren/`

Chapter 4

The Blizzard Challenge Re-evaluation will appear at Interspeech 2021:

J. Taylor and K. Richmond. "Confidence Intervals for ASR-based TTS Evaluation", To appear *Interspeech 2021*. Available: `http://homepages.inf.ed.ac.uk/s1649890/asr/is21.pdf`

Chapter 5

Experiments on representation mixing appeared in:

J. Fong, **J. Taylor**, and S. King. "Testing the Limits of Representation Mixing for Pronunciation Correction in End-to-End Speech Synthesis", in *Proc Interspeech*, 2020, pp. 4019-4023. Available: `http://dx.doi.org/10.21437/Interspeech.2020-2618`

* * *

List of Acronyms

API Application Programming Interface

ARST Attention-based Recurrent Sequence Transducer

ASF Acoustic Space Formulation

ASR Automatic Speech Recognition

BERT Bidirectional Encoder Representations from Transformers

BPE Byte Pair Encoding

CER Character Error Rate

CMU Carnegie Mellon University

CNN Convolutional Neural Network

DNN Deep Neural Network

E2E End-to-End

EATS End-to-End Adversarial Text to Speech

EP Paid Listeners in the Blizzard Challenge, also known as EE.

ER Online Volunteers in the Blizzard Challenge

ES Speech Experts in the Blizzard Challenge

FST Finite-State Transducer

G2P Grapheme-to-Phoneme

GAM General American

GAN Generative Adversarial Network

HMM Hidden Markov Model

IFF Independent Feature Formulation

IPA International Phonetic Alphabet

IV In-Vocabulary

LSA Location Sensitive Attention

LVCSR Large Vocabulary Continuous Speech Recognition

LSTM Long Short Term Memory

NLP Natural Language Processing

NMT Neural Machine Translation

NSW Non-Standard Word

OOV Out-of-Vocabulary

PER Phone Error Rate

POS Part of Speech

RNN Recurrent Neural Network

RP Received Pronunciation

S2S Sequence-to-Sequence

SAMPA Speech Assessment Methods Phonetic Alphabet

SIGMORPHON Special Interest Group on Computational Morphology and Phonology

SPSS Statistical Parametric Speech Synthesis

T2M Text-2-Mel

TTS Text-to-Speech

VAE Variational Auto-Encoder

WER Word Error Rate

WFST Weighted Finite State Transducer

X-SAMPA Extended SAMPA

* * *

Chapter 1

Introduction

1.1 The Pronunciation Lexicon in the Age of End-to-End

In recent years, research in the field of text-to-speech synthesis (TTS) has adopted deep learning with sequence-to-sequence (S2S) models [6] in a drive towards end-to-end (E2E-) TTS. E2E-TTS usually employs a character sequence of either graphemes or phones as input[1]. The authors of Tacotron 2 noted that natural sounding speech ("*difficult to distinguish from human speech*" [7, p.1]) was possible with normalised character sequences only. In [3], the *learned text encoding* from an input character sequence was shown to make linguistic features previously used in hidden Markov model and feedforward deep neural network acoustic models (HMM- and DNN-TTS respectively) redundant. Implicit learning of input character contexts (as was demonstrated in [8]) reduces front-end processing to the stages of text normalisation and an optional grapheme-to-phoneme (G2P) model. S2S modelling also potentially simplifies text normalisation (e.g. [9]), raising the prospect of E2E-TTS from raw text characters.

The shift away from expertly-derived rules and word pronunciations potentially reduces the cost of TTS voice-building. In particular, the shift casts doubt over whether the manual construction of a pronunciation lexicon is required at all for TTS. Prior to the era of E2E-TTS, research on grapheme-based TTS consistently observed decreased performance without the use of linguistic features provided in the front-end [10]–[14]. One such work conducted in Spanish however [13] noted that the relative gap in intelligibility narrowed when increasing the quantity of training data from 1 to 5 hours.

[1]Throughout, the terms *graphemes*, *grapheme-input* and *text-input* are used interchangeably to oppose the use of *phones* and *phone-input* interchangeably.

Given that E2E-TTS makes use of comparatively larger datasets than previously in statistical parametric speech synthesis (SPSS) (e.g. LJ Speech [15] contains 24 hours of audio), the usefulness of a phones instead of graphemes has further been brought into question. In [16] and [5] MOS naturalness scores with non-significant differences were reported with grapheme- and phone-input. Meanwhile, analogous work in E2E-automatic speech recognition (E2E-ASR) has also observed similar word error rate (WER) scores with grapheme- and phone-input [17]–[19].

However, generalisation poses a key challenge in G2P modelling, particularly for words with rare or unusual G2P relations such as foreign words and proper names. Correct pronunciations of foreign words and proper names may be important in a variety of deployed TTS applications (e.g. in smartphone voice interactions). In E2E-TTS with grapheme-input, the G2P model is learnt implicitly from the text-audio pairs in the training set. This thesis explores the performance of implicit G2P performance of E2E-TTS models, with a particular focus on the pronunciation of words of difficult G2P relations. The experiments are mostly conducted in English, but a small experiment is also conducted with an E2E-TTS system in French.

Another recent trend in TTS is intelligibility evaluation by ASR transcription (e.g. as in [20] and [4]). The reliability of using a text-transcription from an ASR system has not previously been analysed in depth. Correlation scores were reported between ASR WER and human transcription WER in [21] and with MOS scores in [22] which indicated a high degree of reliability, but it was unknown whether ASR could reliably rank TTS systems. Using evaluation data collected from 6 years of the Blizzard Challenge, a comparison is conducted between transcriptions by an E2E-ASR system and the Challenge's paid listeners, speech experts, online volunteers. The analysis proceeds to analyse the reliability of E2E-ASR transcription for words of difficult G2P relations. These words are potentially error-prone for both E2E-TTS and E2E-ASR with grapheme-input. Questions thus arise over the empirical fairness of comparing pronunciations by E2E-TTS systems with grapheme- or phone-input with E2E-ASR.

Another perspective on the value of the pronunciation lexicon in English comes from representation mixing in TTS [23]. By mixing grapheme- and phone-input, it becomes possible to specify pronunciations when needed at test time. To train a Tacotron-like model with representation mixing requires phone labels for some of the training data, but it remains unknown how much training data would need to be labelled for a functional phone corrector at test time.

The rest of this chapter provides an overview of background literature, introducing

topics and terms relevant to the reader. A description of the terms phone and phoneme and their use in TTS is followed by a brief overview of how phonetic context is handled in different TTS paradigms. Early work on grapheme-based TTS is analysed and the use of S2S modelling in TTS leads to a discussion about the definition of E2E-TTS. A brief overview of the experiments in this thesis then follows with a list of contributions.

1.2 Definitions

1.2.1 Phone and Phoneme

In this section relevant notions of the terms phone, phoneme and phonetic context will be introduced alongside pronunciation modelling in different TTS paradigms. This background information is provided as context to the reader before proceeding to analyse the use of S2S modelling in TTS, particularly for the E2E-TTS paradigm.

The standards for the representation of speech sounds vary between phonetics, phonology and speech processing for TTS. In phonetics, the most common representation is the International Phonetic Alphabet (IPA) where speech sounds are categorised according to approximate configurations of the vocal tract. Consonant sounds of the IPA are primarily described with voicing (*voiced* or *voiceless*), place of articulation (e.g. *bilalbial*, *alveolar*, *velar*) and manner of articulation (e.g. *plosive*, *fricative*, *approximant*). The primary phonetic attributes for vowel sounds are lip-rounding (*rounded* or *unrounded*), tongue height (*high*/*close*, to *low*/*open*) and tongue advancement (*forward*, *central* or *back*). The IPA can also describe other phonetic attributes including suprasegmentals (syllables, lexical stress), tone levels, tone contours and more fine-grained detail with diacritics.

The IPA phoneset is applicable to all human languages and is a commonly-used standard in phonetics and phonology. In TTS, ASCII encodings of phonesets were originally used and these were typically language-specific. For instance, ARBAbet contains phones of General American English (GAM), and SAMPA contains phones for multiple European languages. X-SAMPA contains ASCII encodings for the entire IPA [24]. With unicode-compatibility, the IPA may also be used directly for example in multi-lingual grapheme-to-phoneme (G2P) modelling [25]. An approximate mapping between phonesets employed in common TTS pronunciation lexica is presented in Table 1.1. Unless stated otherwise, the IPA will be used to refer to pronunciations throughout this thesis, but the TTS systems with phone-input in English used Unisyn

IPA	ARPAbet	Unilex	Combilex	Keyword	IPA	ARPAbet	Unilex	Combilex	Keyword
ə	AX	@	@	COMM**A**	n	N	n	n	**N**AP
æ	AE	a	a	TR**A**P	n̩	EN	n=	n=	GARD**EN**
ɑː	AA	aa	A	P**A**LM	ŋ	NG	ng	N	PA**NG**
aɪ	AY	ai	aI	PR**I**CE	ɔɪ	OY	oi	OI	CH**OI**CE
b	B	b	b	**B**AT	ɒ	OH	Q	Q	L**O**T
ʧ	CH	ch	tS	**CH**AT	ɔː	AO	oo	O	TH**OU**GHT
d	D	d	d	**D**AB	aʊ	AW	ow	aU	M**OU**TH
ð	DH	dh	D	**TH**AT	əʊ	OW	ou	@U	G**OA**T
ɛ	EH	e	E	DR**E**SS	p	P	p	p	**P**AT
ɛɪ	EY	ei	eI	W**AI**ST	ɹ	R	r	r	**R**AT
f	F	f	f	**F**AT	s	S	s	s	**S**AT
g	G	g	g	**G**AP	ʃ	SH	sh	S	**SH**AM
h	HH	h	h	**H**AT	ɾ	T	t	4	BU**T**TER
ɪ	IH	i	I	K**I**T	θ	TH	th	T	MA**TH**
iː	IY	ii	i	FL**EE**CE	ʊ	UH	u	U	F**OO**T
ʤ	JH	jh	dZ	**J**AB	ʌ	AH	uh	V	STR**U**T
k	K	k	k	**C**AT	uː	UW	uu	u	G**OO**SE
l	L	l	l	**L**AD	v	V	v	v	**V**AT
l̩	EL	l=	l=	CATT**LE**	w	W	w	w	**W**AG
ɫ	LW	lw	5	FEE**L**	y	Y	y	j	**Y**AP
m	M	m	m	**M**AT	z	Z	z	z	**Z**AP
m̩	EM	m=	m=	SPAS**M**	ʒ	ZH	zh	Z	BE**IGE**

Table 1.1: Approximate mapping between IPA, ARPAbet (used in the CMU Pronouncing Dictionary or CMUdict) and extensions of SAMPA used in Unisyn and Combilex. Keyword metaphonemes do not map directly to IPA phones since their pronunciation changes depending on dialect. The characters in bold under keywords in this table map to Received Pronunciation (RP).

and Combilex for word pronunciations.

The term *phone* may generally denote a segmental speech sound according to an articulatory description such as in the phonesets above. However, these phones overlie a continuous signal, and thus only approximate phonetic realisations. Details may be added in narrow IPA transcriptions to encompass co-articulation (e.g. assimilation), syllabic, or stress information. Nevertheless, ultimately phones are constrained in their description of speech gestures.

When uttered within a language, a phone may be interpreted as one of the language's *phonemes*. Phonemes describe meaningful speech sounds which determine the minimal pairs of a word. For instance the words *path* and *bath* are meaningfully distinguished by the minimal pair of phonemes: /p/ and /b/. Multiple phonetic realisations may be possible however: for instance the vowel in *bath* or *path* may be realised as [ɑ]

or [a] according to the accent of a speaker. In different accents the concept of *phoneme* may be different. Beyond accent (or geographic/social stratification), phone sounds may also vary according to the surrounding phonetic context (e.g. assimilation), the speech speed (e.g. elision), the interlocutor (e.g. speech convergence/divergence [26]), or a combination of these factors.

Due to the many possible phonetic realisations of phones, the subtleties between the terms *phone* and *phoneme* are complicated. More insight may be gained (in the case of co-articulation) by considering speech sounds as gestures under an Articulatory Phonology approach driven by instrumental methods [27]. Although discrete units only coarsely approximate a phonetic realisation, traditionally in TTS phones are typically used as an intermediary between text-input and an acoustic signal. The phonetic context is then typically learnt from the acoustic signal with the use of features extracted from input text or by encoding an input sequence of characters (text or phones) in an embedding. How phonetic context is handled in TTS depends on the method of acoustic modelling and vocoding which will be covered below.

In Speech Technology the terms *phone* and *phoneme* are often mis-used without consideration for the implied subtleties in their use. In [28], the authors analysed the use of the term *phoneme* in papers accepted to the 2018 Interspeech conference. While 34% of papers contained the word *phoneme*, only 2% gave a definition and 40% misused the term. The authors recommended researchers use the term *phone* unless specifically providing a definition of a phoneme. While phonesets can be used to store contrastive pronunciations of words in a lexicon, lexicon transcriptions for TTS are usually phonetic at the surface-level of a particular accent (e.g. GAM or RP). Furthermore, although TTS models presented do learn phonetic context implicitly, the phones are still used primarily as a phonetic intermediary between text and a TTS acoustic model. Hence, the term *phone* will be used throughout this thesis where other works may refer to *phoneme*, or where in a strict sense the term *phoneme* could correctly be used[2].

In TTS, the pronunciation in output speech depends on how phone sounds are realised in context. In the following sections, methods used in TTS for dealing with phone sounds in different contexts are briefly introduced. First, the phone symbols themselves may implicitly store knowledge about variation depending on character

[2] A parallel could be drawn with the terms *graph* and *grapheme*. In TTS and ASR, the term *grapheme* usually refers to text as it is written in a language (that may also be normalised) rather than to a nuanced distinction from the term graph. Besides the introduction reviewing grapheme-based TTS and ASR, the term *text* is instead generally used to contrast to the term *phone* throughout the thesis.

Keyword	Metaphone	RP	GAM	Examples
M**A**ZDA	aA	a [æ]	A [ɑ]	M**a**rio p**a**sta
BAN**A**NA	Aa	A [ɑ]	a [æ]	Nev**a**da c**a**lve
TR**A**P	a	a [æ]	a [æ]	**A**scot L**a**ngton
P**A**LM	A	A [ɑ]	A [ɑ]	Chic**a**go Gren**a**da

Table 1.2: Example Combilex keyword metaphonemes and their differing realisations in two surface-form accents. Square brackets denote IPA, and all others are Combilex symbols (X-SAMPA).

contexts as with metaphonemes.

1.2.2 Metaphoneme

Dialect variation may be modelled using phoneme-2-phoneme (P2P) transformations across accents [29]–[31]. These approaches typically rely on transforming phones from one accent or dialect to another. However, at a deeper level phonetic variation based on speaker accent in English may be recorded with keyword phonology. Described in [32] and adopted to Unisyn in [33], [34], keyword pronunciation lexica employ an underlying baseform lexicon where word pronunciations are stored in metaphonemes which can then be transformed to surface-form accents.

A metaphoneme is denoted by a representative keyword. Looking at Table 1.2, it is said that the 'the M**A**ZDA vowel' refers to the metaphoneme 'aA'. Words containing the M**A**ZDA vowel, such as *Mario* and *pasta*, are transformed to [æ] in RP and to [ɑ] in GAM. Note that the transformations are not identical for the M**A**ZDA, BAN**A**NA, TR**A**P and P**A**LM vowels. Importantly, these denote distinct sets of vowels which have differing transformations in the RP and GAM accents. For instance the transformation is reversed for the BAN**A**NA vowel with [ɑ] in RP and to [æ] in GAM.

Word entries in the baseform lexicon are written in metaphonemes. Metaphonemes abstract away from phonetic realisations of keywords observed across different ac-

cents. From metaphonemes, context-dependent transformation rules are used to create new surface-form lexica in different accents. Creating a lexicon this way aimed to avoid the effort that would be incurred by building/maintaining separate lexica for different accents. However, writing metaphone-to-phone transformations may require specialist knowledge of a given target accent which is still laborious even if more efficient than building new lexica from scratch.

Usually, speech recordings used in single-speaker TTS have a GAM accent (e.g. LJ Speech or Nancy - see Table 2.1) for which surface-form phones are readily available in open sourced tools such as the CMUdict. The pronunciation corresponding to surface-form phones will depend on how a TTS system models phone sounds in context. Phonetic context in acoustic modelling varies depending on the TTS paradigm in question (see Section 1.2.3 below).

Important to note however is that metaphonemes store variation which cannot be learnt from the surrounding phonetic context alone (such as the knowledge of the differing transformations of the M**A**ZDA and BAN**A**NA vowels above). In this thesis, further examples where pronunciations may not be predictable from surrounding phonetic or character context will also be analysed.

1.2.3 Phonetic Context in Acoustic Modelling

Phones are limited in the phonetic detail they describe in speech signals. As discrete symbolic representations they do not describe co-articulation - especially the simple phonesets used in TTS. For example the pronunciation of *handbag* may vary: the voiced alveolar plosive /d/ can be deleted and the voiced alveolar nasal /n/ often assimilates the bilabial place of articulation of the following phoneme /b/: [hæ**mb**æg]. While some phonetic variation based on context can be designed into a pronunciation lexicon (see Section 1.2.2 above), phonetic variation based on context is typically missing from the phonestrings in a pronunciation lexicon in TTS. Instead, phonetic context is handled in different ways depending on the TTS pipeline employed as will be explained below. Figure 1.3 presents a diagram of how phonetic contexts are handled in unit selection, HMM-TTS, DNN-TTS and E2E-TTS.

1.2.3.1 Unit Selection

In unit selection for instance, the units of natural speech from a database are concatenated together. The unit of choice spans an arbitrary length that typically crosses

phone boundaries (diphones, triphones or quinphones). The span of the unit, the selection of appropriate units and the minimisation of join mismatches together determine the audio output that represent phonetic contexts in unit selection. The Independent Feature Formulation (IFF - [35]) labels units with linguistic metadata such as neighbouring phones, syllabic, word and phrasal information, and POS tags. These labels come from a series of modules collectively known as the front-end. The IFF may be used to calculate the target cost of candidate units. The join cost would minimise audible artefacts between units by measuring mismatches in F0, energy and spectral mismatch. The acoustic space formulation (ASF) selects units according to acoustic similarity. Phonetic context in unit selection is thus handled by supplemental linguistic and or acoustic analysis which drives retrieval and concatenation of units. The acoustic realisation of *handbag* would depend on the span of the unit (e.g. diphone), processes of unit retrieval, concatenation and smoothing.

1.2.3.2 Phonetic Context in SPSS

In statistical parametric speech synthesis (SPSS), acoustic models predict vocoder parameters given contextual linguistic features. HMM-TTS employs the front-end to provide questions in a decision tree. The leaves of the decision tree map the linguistic contexts to an acoustic space to generate the speech parameters fed to acoustic/ duration models and a vocoder.

However, decision trees are ultimately limited. The number of definable contexts becomes exponential and no amount of training data could comprehensively provide a mapping between the linguistic features and the acoustic parameters [36]. DNN-TTS employs feed forward (and later recurrent) neural networks to learn a distributed representation of contextual features from frame-wise one-hot vectors which are embedded into a high-dimensional space. The distributed representation within DNNs replaces the feature extraction of decision trees.

In SPSS thus, the pronunciation of *handbag* in context is learnt via regression of defined linguistic contexts and speech parameters. The linguistic contexts are once again provided by the TTS front-end. In [8], the most important features in DNN-TTS were investigated. Based on results in [37] and [38], these features are listed in Table 1.3.

With the adoption of S2S models in acoustic modelling, work towards E2E-TTS replaced explicit context features with character sequences [3]. Importantly, the implicit learning of context raises the question of how front-end analysis is conducted

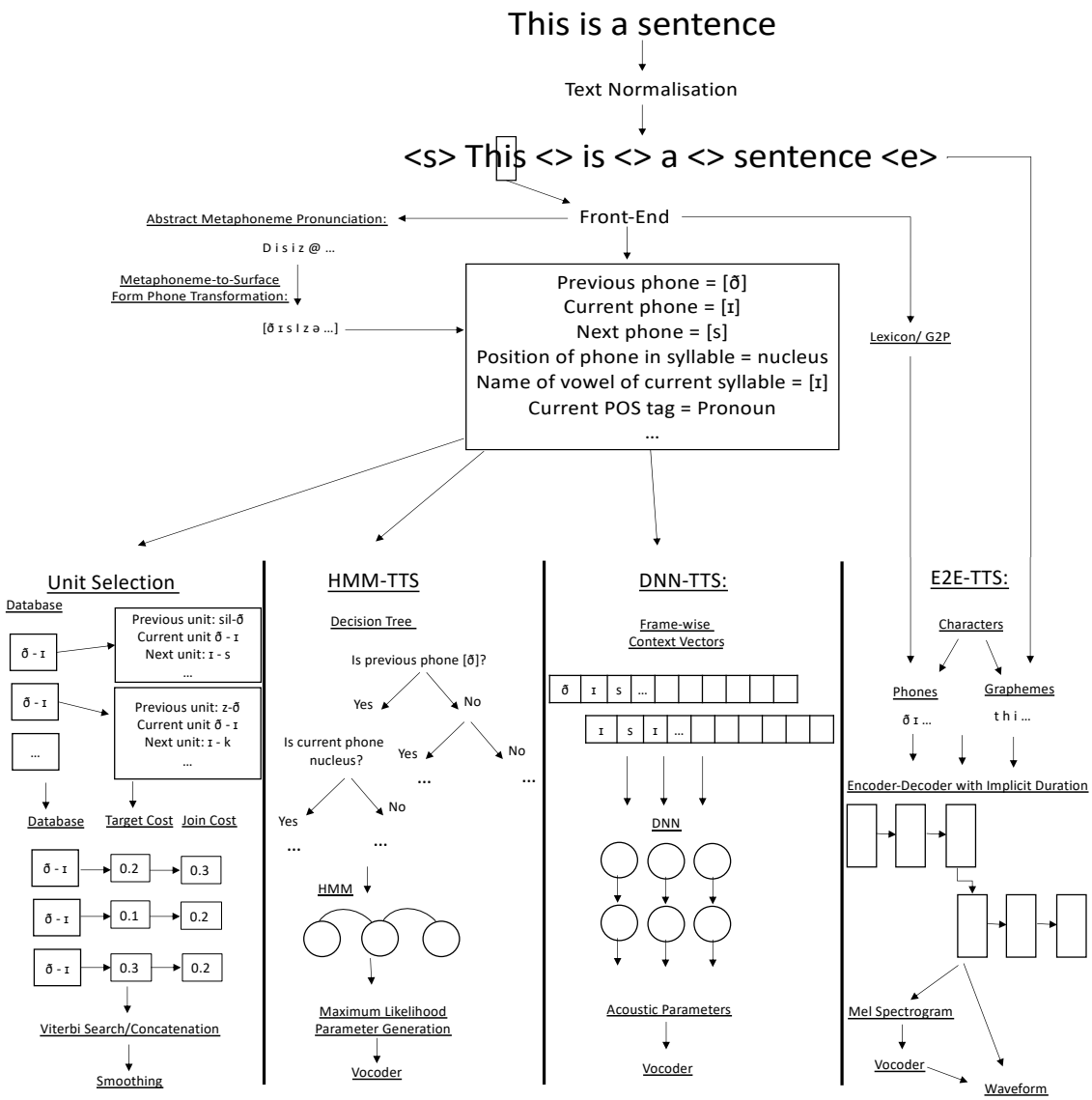

Figure 1.1: Diagram of phonetic context features in different TTS paradigms. The front-end provides context features for unit selection, HMM-TTS and DNN-TTS. E2E-TTS takes graphemes, phones or a mixture as input usually without additional features.

Linguistic Level	Feature in unit selection	Important Feature in SPSS
Phonetic Context	- Left/right phonetic contexts	- Previous, current and next phone identities
Syllable	- Position in syllable (initial, medial, final and inter-word)	- Whether current syllable is stressed or not
		- Name of vowel in current syllable
Word	- Position in word (initial, medial, final and inter-word)	- Position of current syllable in word
	- Stress is correct	
POS	- POS is correct	- POS of current word
Phrase	- Position in phrase \\(initial, medial or final)\end{tabular}	n.a

Table 1.3: Context Features in Multisyn (unit selection [39]) and the most important features in SPSS. In Multisyn, each context feature receives a weight in the calculation of the target cost. The features are applied to units such as diphones. The features presented for SPSS are a subset calculated to be the most beneficial in [37]. The subset of important features are referenced in Section 1.5.1 when discussing context feature learning in E2E-TTS from an analysis in [8]. A comprehensive set of context features used in SPSS is provided in [40].

and whether text-input may be used instead of complex analyses in the front-end. In the next section, grapheme-based TTS in traditional TTS paradigms will be covered before introducing S2S models and analysing implicit phonetic context learning in E2E-TTS.

1.3 Grapheme-based TTS

Traditionally, the modules of the front-end have required expert linguistic knowledge to label text-input at different linguistic levels (as shown in Figure 1.3). The cost of front-end processing is considerable, especially to build text-normalisation rules and a pronunciation lexicon. For low resource languages, the required expertise for front-end processing may not exist or arguably may not be necessary with a regularly phonetic orthography. The constraints to create linguistic resources motivated research into grapheme-based TTS.

The first use of grapheme-based TTS in a unit selection system was [10]. A unit-selection system was also used in [11]. Grapheme-based TTS was also explored for decision-tree/HMM-TTS for English in [12], for Spanish in [13] and for 12 languages in [14]. Importantly, surrounding context still needed to be defined for unit selection and HMM-based TTS. In these works, grapheme-based systems underperformed their counterparts that used phones and contextual features.

In [11], the intelligibility of using graphemes in unit selection was improved by

naively expanding grapheme-models to tri-grapheme models. In the works with HMMs, intelligibility was lower in English and Spanish. In Spanish, the system had difficulties learning two-letter-to-one phone mapping. For example the letter *h* in Spanish is usually silent but was confused with [tʃ] as in words like *churro*. Monosyllabic words were also frequently omitted such as *hay*, *un* and *la*. However, it was also found that the gap in intelligibility between grapheme- and phone-input narrowed from 12% to 8% on Spanish Harvard test stimuli when training data was increased from 1 to 5 hours. Nevertheless, all phone-based systems (including those trained on 1 hour of data) performed with a WER of 4% or below. In English, larger gaps in performance were observed. For instance the best performing grapheme-based system in [12] scored 42% WER compared to 25% from the phone-based system.

In [14], grapheme-based HMM baselines (built with clustergen [41]) in 12 languages were compared with phone-based systems employing a universal phonetic transcription method known as Unitran [42]. In the results, there was a clear preference for systems with Unitran or linguistic knowledge (including phones where available) over the grapheme-based baselines. However to what extent are word pronunciations improved in grapheme-based TTS with the contextual learning benefits of S2S models trained on larger amounts of data?

1.3.1 Grapheme-based ASR

In ASR, it was observed that grapheme-based HMM systems performed with very similar WERs to phone-based equivalents when increasing the amount of data to 1200 hours in [43], and these results have been echoed with S2S models with even less data. With the 300 hour Switchboard corpus in [18] the performance of the grapheme-based system was only negligibly below the phone-based system. With the 960 hour Librispeech dataset the grapheme-based system outperformed the phone-based system in terms of WER on the clean and other test sets. These findings were corroborated in [19], and similar findings were shown when approximately 12,500 hours of speech were used to train models in [17]. Given the similarity in WER performance in ASR models with grapheme- and phone-input, it is natural to ask to what extent does grapheme- or phone-input matter for S2S TTS acoustic models?

More broadly, S2S models can simplify text normalisation and G2P modelling to enable E2E-TTS from raw text characters. Thus in the next section S2S modelling for modules in the TTS pipeline will be reviewed to cover how necessary linguistic

analyses are in TTS in English given the advances of deep learning with large amounts of data.

1.4 Sequence-to-Sequence Modelling

In recent years, a large amount of research has studied the application of S2S models in TTS, as in related fields such as ASR, Natural Language Processing (NLP), Machine Translation (MT). Simultaneously, research has sought to simplify the TTS pipeline towards E2E-TTS. In this section, the use of deep learning in front-end modules and acoustic modelling will be reviewed.

1.4.1 The Front-End

The TTS front-end consists of a pipeline to normalise input text and generate a linguistic specification for use by duration, acoustic, prosody and vocoder models. Modules in the front-end pipeline for English typically include tokenisation, non-standard word disambiguation, part-of-speech (POS) tagging, a pronunciation lexicon, letter-to-sound (LTS) or G2P prediction for out-of-vocabulary (OOV) words and post-lexical processing.

Traditionally, each module requires a separate processing step. Since a key topic in E2E-TTS for this thesis is the question between grapheme- or phone-input, here text normalisation (also known as TN) and G2P modelling will be reviewed specifically.

1.4.1.1 Text Normalistion: Tokenization and Verbalisation

Tokenisation classifies tokens of text-input. Examples of token classes include: ordinary words, punctuation, individual or unknown characters (such as emojis) and non-standard semiotic classes or words. Non-standard semiotic classes or words (sometimes known as NSWs) include measures (*km, cm*), currency symbols (*$, £*, dates (of varying forms: *d/mm/yyyy*, *m/dd/yy*) ordinal numbers (1st, 2nd, etc), cardinal numbers (*one*, *two*), fractions, times, telephone numbers and addresses.

After classification the tokens are *verbalised*. Verbalisation uses language-specific rules to disambiguate and expand identified tokens into a spoken form - see [44] or [45] for more information. Manual creation of rules for tokenisation - especially of non-standard classes - makes front-end development an expensive process.

Tokenization and verbalisation can be treated as a neural machine translation (NMT) task where the source is unnormalised text and the target normalised text or phones in a spoken form. In [9], the authors used recurrent neural networks (RNNs), observing high accuracy but unacceptable errors for TTS in deployment. The authors ran a Kaggle challenge to improve RNN-based text normalisation and avoid the errors in English, Polish and Russian. Since this time, proposals to improve neural text normalisation with S2S models have included convolutional neural networks (CNNs) [46], [47] and transformers [48] for faster training and inference. In [49], the authors proposed decomposition of input text into subword units and additional linguistic features derivable from text-input.

In Mandarin, text normalisation is also an important task. Word segmentation identifies logographs in context to meaningfully disambiguate words into a spoken form [50]–[52]. In [53], the authors proposed a hybrid approach leveraging multi-head self-attention with hand-written text normalisation rules. S2S models have also been proposed to jointly perform front-end tasks in Mandarin such as polyphone disambiguation, POS tagging and prosody prediction [54] and in 19 languages in [55]. However, in the results of the *Text Normalisation with RNNs* challenge, the best performing systems still heavily relied on manual rules [56]. Text normalisation using S2S models still produce irrecoverable errors which post-filters cannot fix such as transcribing *but* as *Sunday* [48].

Another obstacle in their deployment is the amount of labelled data needed to perform supervised learning for new languages. A large public dataset exists for English (approximately 3.6GB released for the aforementioned challenge [57]), but labelling data for text normalisation is an expensive process. In [58], the authors proposed to perform S2S text normalisation with less labelled data using a granular tokenisation method. Prior to S2S modelling, an unsupervised approach to front-end processing was proposed in [59]. Tools dedicated to unsupervised text normalisation were also developed during the *Simple4all* project [60].

Text normalisation with at least some manual rules is the most reliable approach in practice. Recent work has thus sought to productively extract relevant tokenisation and verbalisation rules from speakers of low-resource languages. In [61], the authors build finite state transducer (FST) grammars for Bangla, Khmer, Nepali, Javanese, Sinhala and provide advice for working with linguists. Text normalisation for Burmese was also created in [62]. A rapid identification and creation of FST-rules for new languages using a template-based questionnaire was also proposed in [63]. The reliance of text

normalisation on manual rules poses an obstacle for E2E-TTS from raw-text input. Research on E2E-TTS acoustic modelling usually employs pre-normalised text.

1.4.1.2 Lexicon and G2P

After the verbalisation of input text, a phonetic representation is usually predicted. Word pronunciations are stored in a lexicon if they are available or predicted from a G2P model if the words are out-of-vocabulary (OOV - missing from the lexicon).

Inconveniently, manual creation of a pronunciation lexicon is laborious and expensive. Aside from accurate phone strings (potentially in multiple accents), pronunciation lexica may also contain more linguistic metadata stored as fields in a database. Unisyn and Combilex contain metaphonemes, syllabic and morphological boundaries, POS tags and lexical stress for word entries. Due to the finite nature of pronunciation lexica, a G2P model is always necessary for TTS in deployment.

G2P can be rule-based (LTS) or data-driven. Rule-based LTS typically encodes rules in an FST with an accompanying exception list. Data driven G2P by contrast aims to learn the regular pronunciation rules of a language from examples in a lexicon. As outlined in [64], two key aims of data-driven G2P are lexicon compression (learning from as small and as regular a lexicon as possible) and generalisation (predicing pronunciations for unseen words). Compression and generalisation are particularly difficult in English due to its complex LTS rules (see Figure 2.2).

Data-driven methods for G2P include local classification (or 1-to-N alignments typically with decision trees - [65]–[71]), pronunciation by analogy [72]–[75] and joint-sequence models [76], [77]. The approach employed in the commonly used *Sequitur* package [78] is based on joint-sequence models where aligned graphemes and phones are learnt as joint tokens or graphones [64]. Proposed improvements to graphone models with language model rescoring include [79] and [80]. For more information on data driven G2P methods prior to S2S modelling, see [64].

S2S models applied to G2P include RNNs with long short term memory (LSTM) units[3] [81], [82], CNNs [83], [84] and the transformer [85]. Augmenting S2S G2P model with analogy information from a lexicon was proposed in [86]. In [87], syllable boundaries and stress markers were jointly predicted in an RNN, where improvements in G2P prediction were found for some languages. Neural models have also been

[3]Note, the meaning of *unit* here is a cell in a neural network. This is different from the meaning of *unit* in *unit selection* which refers to an audio sample that spans an arbitrary length of phone sounds (e.g. diphone, triphone or quinphone). A separate meaning of *unit* is also used in Section 3.3 where it refers to either words or morphemes.

applied to multilingual G2P [25], [88], [89] which was the subject of the 2020 and 2021 *SIGMORPHON* challenge [90].

1.4.1.3 Pronunciation of Proper Names and Foreign Words

Despite advances in G2P with S2S modelling, pronunciation of proper names and words of foreign origin remain a challenge. Learning word pronunciations for proper names and foreign words has typically focused on G2P for ASR [91]–[94]. Early work on G2P for names employed language origin information - as in [95], [96]. Approaches where pronunciations are gathered via online users include [97] and through Crowdsourcing in [98]. The use of an ASR acoustic model has been investigated to improve the recognition of names with Google voice search [99]–[101]. It is not clear how reliable these methods could be for correct word pronunciations in TTS. A similar area of research on the pronunciation of names is nativization [102], [103]. Foreign words (including names) may possess G2P relations not typically used in a language. Nativization is the adoption or adaption of foreign G2P relations into a the native language of a TTS system.

1.4.2 Acoustic Modelling

A general development of TTS acoustic models in recent years is presented in Figure 1.2. As mentioned above, two key approaches to acoustic modelling in TTS are unit selection or SPSS. Both of these approaches make use of front-end text analysis. In unit selection, concatenation of candidate units generate a waveform. Units are selected either according to an IFF where linguistic features specify similarity of candidate units, or according to an acoustic similarity under an ASF. IFF features include phones (and phonetic class information - manner, place, voicing etc), syllabic, stress and POS tag information. The ASF includes features such as F0, spectral energy or amplitude, and spectral mismatch. In [104], 10 years of TTS systems up to 2014 were analysed with the author concluding unit selection systems were more natural and SPSS systems were more intelligible but robotic. The ASF employed models used in SPSS, which led to a cross-over known as *hybrid* TTS. Examples of these include unit selection driven by DNNs [105] and S2S models [106].

RNNs with LSTM units were first proposed as S2S models for SPSS in works such as [107], [108]. These can still be considered an extension of DNN-TTS since they use aligned linguistic context features as input. In [3], it was shown that the use of a phone

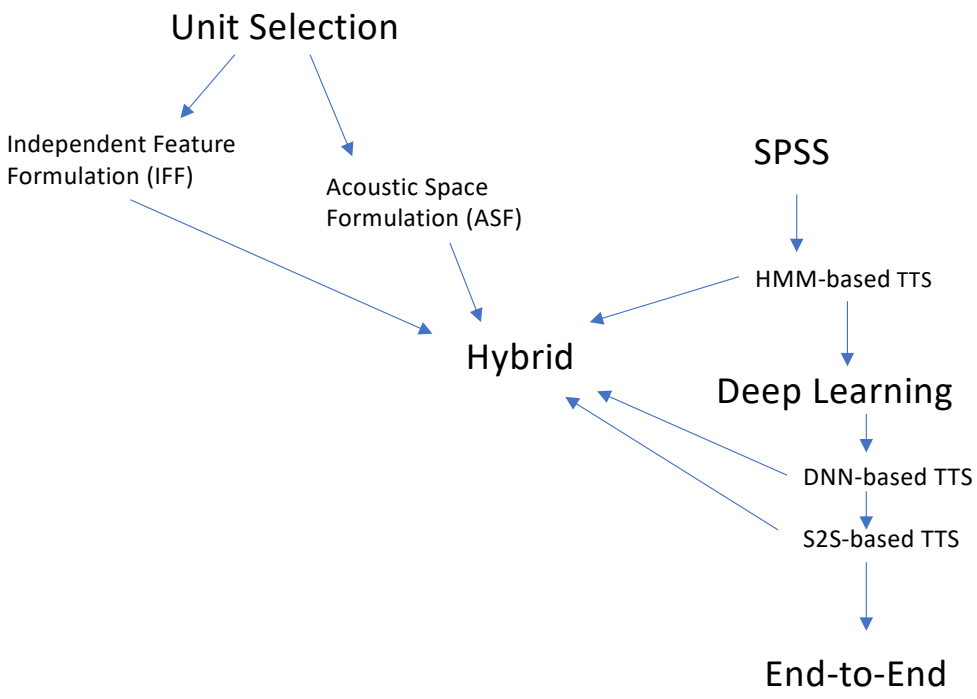

Figure 1.2: Paradigm shifts in TTS from unit selection to E2E-TTS.

character sequence (the *learned text encoder*) instead of linguistic context features improved naturalness. An input sequence of characters is an attribute of E2E-TTS, described further below.

1.5 E2E-TTS

E2E-TTS was first defined in [109]:

"*The term "end-to-end" means that text analysis and acoustic modelling are accomplished together by an attention-based recurrent network, which has the capacity to learn the relevant contextual factors embedded in the text sequence.*" - [109, p.1]

Whereas in (feedforward) DNNs, context is provided by a frame-wise vector of contextual linguistic features, S2S models learn a contextual representation from an input sequence of characters. Whereas in DNN-TTS the duration of a phone was modelled determined by a Viterb-based aligner, in [109] (also known as the "*attention-based recurrent sequence transducer*" - ARST), the HMM aligner drove a neural attention mechanism of an LSTM acoustic model.

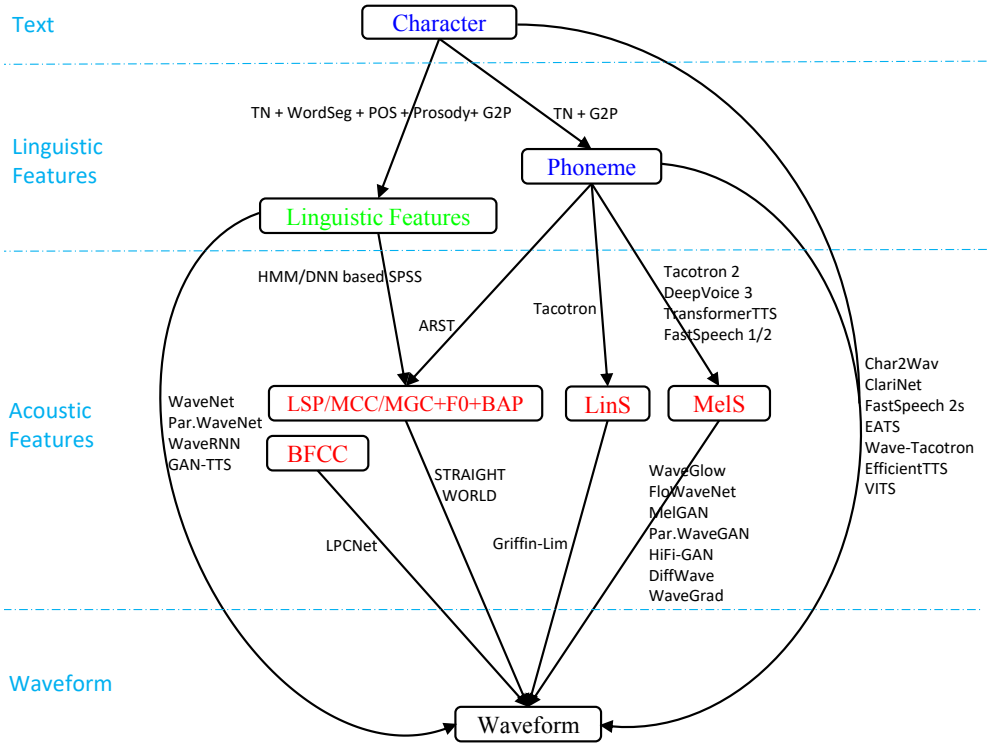

Figure 1.3: Pipelines of E2E-TTS from [110]. Traditionally the front-end predicts linguistic features and an acoustic model predicts vocoder parameters which are used by a vocoder to produce a waveform. In E2E-TTS, the front-end normalises input text and may also predict phones. In Tacotron the acoustic model predicted a linear spectrogram, but Tacotron 2 and Deep Voice 3 [111] (and DC-TTS [112] used in Chapter 2) predict a mel spectrogram which is then converted to a waveform via neural vocoders such as WaveRNN ([113] used in Chapter 3), WaveGlow [114], FloWaveNet [115], MelGAN [116], Parallel WaveGAN [117], HIFI-GAN [118], DiffWave [119] and WaveGrad [120]. Since these can be trained on predicted mel spectrograms from an acoustic model, they can be considered E2E-TTS despite separate training of the vocoders. The only fully E2E-TTS compatible models (that run from characters to waveform) without separate training are FastSpeech 2s [121], EATS [20], Wave-Tacotron [4], Efficient-TTS [122] and VITS [123], since Char2Wav [1], Clarinet [124] require extra steps in training. In this thesis, when S2S acoustic models use graphemes/text-input they will be referred to as E2E-TTS despite the use of Griffin-Lim or vocoder to obtain waveforms. Some neural vocoders can produce a waveform from linguistic features such as WaveNet [125], WaveRNN [113], Parallel WaveNet [126] and GAN-TTS [127] but the reliance of features precludes this pipeline from the classs of E2E models. The STRAIGHT and WORLD and LPCNET vocoders [128]–[130] were used in HMM-based and DNN-based pipelines, using acoustic parameters to model a waveform. In this thesis, when S2S acoustic models use graphemes/text-input they will be referred to as E2E-TTS despite the use of a vocoder to obtain waveforms. This figure has been reproduced with permission from the original authors.

The Char2Wav [1] system proposed an E2E system using text-input (without a front-end) with an attention-based S2S acoustic model. The aligner was also replaced by attention in Tacotron [2], which predicted a spectrogram. Figure 1.3 describes the aim of E2E-TTS with Tacotron in more detail. One of the key aims of E2E-TTS outlined in early works was to replace the lexicon and G2P model with an encoder.

E2E-TTS has since employed a variety of architectures. In summary, some trends in E2E-TTS have included the adoption of CNNs [7], [111], [112] and transformers [131]; the replacement of autoregressive decoding with multi-head attention or depthwise convolutions [132] for faster decoding; the replacement of brittle soft attention mechanisms with hard monotonic attentions [16], [133]; the replacement of attention with alternative, implicit duration models such as VAE [134]; unsupervised duration models [135] for expressive or controllable TTS, also with pre-trainined language model embeddings such as BERT [136], [137]; multi-speaker TTS [138], [139]; multi-lingual TTS and cross language voice cloning [140], [141]; low-resource TTS [142] and streaming, or real-time TTS [143]–[146]. For a wide-randing survey on neural models in TTS see [110]. For a review of deep learning in E2E-TTS see [121].

Arranging the large number of works on TTS with deep learning in recent years is not straightforward. According to [110], alternative taxonomies for the work on neural TTS include autoregressive v non-autoregressive, generative model types (S2S, Flow, GAN, VAE) and network structure (CNN, RNN, self-attention and hybrid). See Section 2.6 of [110] for more details.

Given the variety of work and approaches, the definition of E2E-TTS has become ambiguous and multi-faceted. Early-on, it was proposed in [147] that the use of neural sequence models for each stage of the SPSS TTS pipeline was E2E (a pipeline of individually trained networks e.g. a S2S G2P model, a S2S text-encoder, a S2S vocoder). However, E2E-TTS usually implies a pipeline that does not require training of separate modules. Some early key attributes of E2E-TTS included the use of: 1) raw text/ minimal normalised text, 2) S2S encoder-decoder models with attention, 3) implicit vocoding. Over-time the attention mechanism in point 2) was shown to be redundant [3] and could lead to unrobust acoustic modelling. The E2E-TTS attribute of an attention mechanism could more broadly be understood as a method for implicit duration modelling as in [135]. In this thesis, DC-TTS and Tacotron are used which employ a CNN and a CBHG encoder respectively to predict mel spectrograms. The mel spectrograms are then converted to waveforms with neural vocoders. These neural vocoders must be trained separately from the text encoders. This contrasts to the class of E2E-

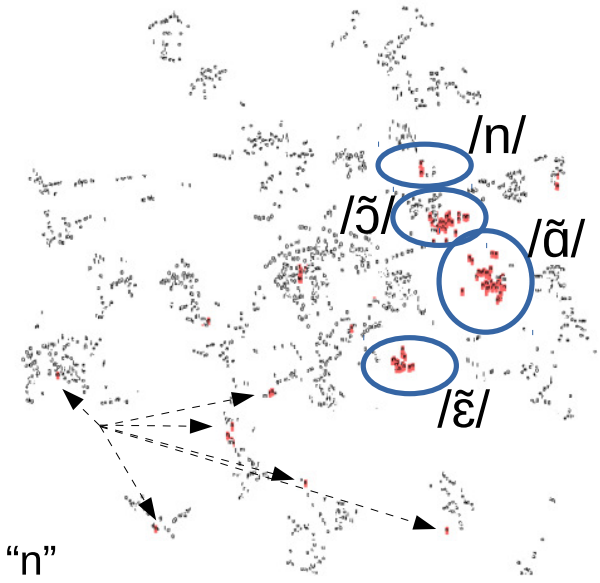

Figure 1.4: T-SNE visualisation of grapheme ’n’ in Tacotron trained on French data from [5]. The authors highlighted the embeddings of the grapheme in red and reported clustering of different nasal phone identities depending on context. However, clustering of grapheme-contexts does not indicate that pronunciations of words - especially words with unusual G2P relations - are robustly modelled.

TTS models which convert directly from text to waveform (which will be referred to as *single-stage* or *fully* E2E-TTS). The difference is illustrated in Figure 1.3

1.5.1 Phonetic Context in E2E-TTS

A key difficulty when working with neural networks is the black-box which makes interpretability of learnt knowledge more difficult than when using the IFF in unit selection or decision trees in SPSS. The replacement of contextual features in a feed-forward DNN with a S2S text-encoder consuming a character sequence was observed to lead to significant improvements in naturalness in [3]. However, questions still remain when using only character sequences. How much phonetic context is implicitly learnt? Should the character sequence be composed of graphemes or phones?

The first question was tackled in [8], where the embeddings of a Deep Voice 3 system were treated as a pre-trained model for a classification task of the most important linguistic context features in Figure 1.3. The E2E-TTS model was trained on phone-input from a G2P model using the ARPAbet. The classification task involved predicting each feature given the input.

The accuracy of classification of previous, current and next phonemes was 73.1%, 84% and 67.1% respectively. While overall these are high classification scores, it was noted that accuracy was particularly difficult for vowel identities which vary a lot in English depending on context and stress. However, when predicting POS-tags, words spelt similarly would be confused (such as *wood* as a noun and *wooden* as an adjective). This suggests that E2E-TTS models may not be learning a deep linguistic structure of language.

Another work which has attempted to understand phonetic contexts in E2E-TTS is [5]. The authors used t-SNE [148] to observe embeddings of input graphemes, finding that they mapped to different acoustic spaces depending on the grapheme context. For example the grapheme *n* in French is [œ̃] in a word like *un* but [ɑ̃] in word like *banque*. This visualisation demonstrates phonetic contexts can be learnt in E2E-TTS. However, obtaining a correct word pronunciation may require deeper knowledge than is possible in sentence level text-audio pairs. A G2P model whether implicit (using text-input) or explicit (trained separately to predict phones) may be insufficient for TTS in deployment. This point will be argued in further detail throughout this thesis.

1.5.2 Grapheme- or Phone-input for E2E-TTS?

Early work on E2E-TTS (e.g. Tacotron [2]) claimed that the lexicon as well as the G2P model could be replaced by a neural text encoder. Word pronunciations are then modelled using only the data available in the training data. This approach does not use a pronunciation lexicon, which bypasses a considerable cost in building front-end resources for TTS in new languages. This approach to TTS is part of a more general trend in speech processing to move away from manual resources for tasks such as text normalisation and acoustic modelling in ASR: where the relationship between input and output sequences can be learnt with a character representation of either text or phones.

The use of a front-end in traditional paradigms of TTS would include text normalisation before phonetisation. In a strict sense, E2E-TTS should use (unnormalised) text-input but to date, text-input in E2E-TTS models still use some form of normalised text-input. The first Tacotron paper [2] employed normalised text-input to an encoder (see 1.5). The authors noted that some text normalisation was necessary but that neural text normalisation as in [9] could simplify the process. Importantly for E2E-TTS however: reliable text normalisation is not simple to build. Text normalisation with S2S

models require labelling a large amount of non-standard words and the models still make unacceptable and irrecoverable errors [48]. S2S text normalisation also requires explicit training. At the time of writing, subsequent Tacotron papers (including the most recent [4], [149]) still employ Google's proprietary text normalisation which is reliant upon manual rules. Text normalisation therefore presents a challenge to building fully E2E-TTS systems.

If phone-input is employed in E2E-TTS, the quality of phone-labels are often unknown. For instance, the rule-based LTS module from E-speak [150] and online S2S-based G2P models found on GitHub [151], [152] are commonly used for the E2E-TTS voice building recipes in the collaborative ESPnet-TTS [153] toolkit. In English, the use of phones in E2E-TTS therefore implies G2P predictions rather than gold-standard labels from a lexicon. This thesis compares the pronunciation of E2E-TTS to gold-standard phone labels, to understand whether there is still a benefit to using a pronunciation lexicon for phone-strings instead of a G2P model.

1.6 Thesis Layout

The starting point in Chapter 2 is to measure the quality of G2P models trained with only the words contained in E2E-TTS datasets. The chapter proceeds to evaluate the pronunciation of words with difficult G2P relations and the quality of grapheme-input to an early E2E-TTS system based on CNNs: DC-TTS.

Following the observation that pronunciations in DC-TTS made generalisation errors with words that contain difficult-to-model G2P relations[4], in Chapter 3 the use of morphological boundaries (which offer an augmentation to phones) is examined to improve pronunciation modelling. In the results, it is found that the pronunciations of some words require more knowledge than can be learnt from surrounding context when using text-input. This is explored further with an evaluation of implicit pronunciation modelling in French with more examples of difficult G2P relations and *liaison*. These experiments are conducted with an implementation of Tacotron.

In Chapter 4 the thesis analyses another recent trend in TTS: the use of E2E-ASR as a proxy metric of intelligibility. The chapter begins with a re-evaluation of 6 years of results from the Blizzard Challenge to inform of the reliability of this approach in

[4]What is meant by these words will become clear in Section 2.5.2. Briefly, they are a set of words that were mispronounced by a S2S-G2P model. These will usually be referred to as *words of difficult G2P relations*, *difficult G2P words* or *G2P error words*. In Chapters 4 and 5 the set of words is known as *Out-LJ*.

ranking the relative performance of TTS systems. Since pronunciation evaluation in preceding chapters are on a small-scale and subjective, the reliability of E2E-ASR transcription for difficult G2P words is also investigated. Since E2E-ASR only provides grapheme-based transcription, an in-depth analysis is conducted to explore whether this form of transcription can be appropriate to evaluate pronunciation modelling in grapheme- or phone-input to E2E-TTS.

The use of grapheme- and phone-input is investigated further in E2E-TTS with an experiment in representation mixing in Chapter 5. The idea behind representation mixing is to use phones as an alternative to graphemes when the latter presents ambiguous G2P relations. Varying amounts of training data are phonetised in an attempt to find the smallest pronunciation lexicon necessary to enable pronunciation correction. Results from a small-scale subjective listening test are compared with E2E-ASR transcription.

In Chapter 6, the thesis concludes with an analysis of the value of the pronunciation lexicon in E2E-TTS. Some concluding remarks aim to offer insight in anticipation of lingering questions the reader may have, with proposals for future research in the comparison of text- and phone-input in E2E-TTS.

1.7 Contributions

In the following chapters, the performance of E2E-TTS is analysed with text- and phone-input. Original contributions include:

Chapter 2

Modelling word pronunciations is explored in E2E-TTS via a simulation using G2P models. Implicit G2P models learnt from E2E-TTS data are shown to score higher error rates than the same G2P model trained on a pronunciation lexicon in English. An accompanying analysis of G2P and TTS model errors is provided. Text- and phone-input is compared in DC-TTS in an attempt to find an approximate phonetic error rate of text-input with words of irregular pronunciation in English.

Chapter 3

Following the observation that some word mispronunciations occur around morpheme boundaries (e.g. *th* in the word *pothole* was pronounced as [ð]), the use of morphological boundaries is investigated in a S2S-G2P model and in an implementation of

Tacotron trained on text-input in English. The further observation that some word pronunciations are difficult to predict from surrounding phonetic context alone (e.g. such as names like *Siobahn*) inspires preference tests with stimuli containing difficult to pronounce G2P words in French. These experiments follow the findings of [5] where it was suggested there was little difference between text- and phone-input to Tacotron in French. Specifically, G2P error words and post-lexical cases of disallowed *liaison* in French are found to be preferred by phone-based systems with gold-standard phone strings.

Chapter 4

In an attempt to conduct a larger-scale evaluation of TTS output, an exploration of ASR as an intelligbility metric is conducted. The reliability of rankings of systems in the Blizzard Challenge is studied. An analysis of E2E-ASR transcription to approximately measure pronunciation modelling in E2E-TTS is conducted. While on aggregate E2E-ASR is found to be more reliable than the non-native listeners in the Blizzard Challenge, specifically for words of difficult G2P E2E-ASR is shown to be biased in the transcription of words with difficult G2P (or P2G). Notwithstanding the imperfections in E2E-ASR transcription, E2E-ASR detects mispronunciations by TTS systems and text- and phone-input are compared on a large scale.

Chapter 5

Experiments measuring pronunciation correction using representation mixing are conducted. Text-input is mixed with phone-input from lexica of a varying number of n word types. Results from a small-scale listening test are compared with a large scale evaluation of intelligibility using E2E-ASR. Clear improvements in pronunciations are shown when adopting representation mixing with phones.

Chapter 6

The following findings are submitted as a thesis:

- Pronunciation control is desirable in certain deployable TTS applications to ensure correct pronunciation.
- The pronunciation of some words cannot be predicted by generalising from surrounding character contexts alone. To accurately guide the correct pronunciation

of these words requires prior knowledge as represented in a lexicon.

- E2E-ASR transcription can be more reliable than transcription by unreliable judgements from untrained listeners to approximately evaluate the intelligibility of TTS systems. However, E2E-TTS transcription can be unreliable for words of difficult G2P relations.

Some concluding remarks address the value of the pronunciation lexicon in TTS whilst providing further information in anticipation of questions the reader may have. Proposals for future work are also presented.

1.8 Summary

In this chapter, pronunciation modelling across TTS paradigms was introduced. S2S modelling for text normalistion, G2P and acoustic modelling was introduced in the context of work towards E2E-TTS. A discussion about the use of text-input in E2E-TTS and the value of the pronunciation lexicon motivated research questions which are tackled in the next chapter. A list of contributions was also presented.

Chapter 2

Implicit Pronunciation Modelling in E2E-TTS

As discussed in the previous chapter, in the stricted sense E2E-TTS uses S2S text encoders without extensive front-end processing. Under E2E-TTS, the G2P model usually becomes implicit when learning acoustic representation from an input sequence of text. Questions arise about the reliability of E2E-TTS pronunciations with the substitution of a manually-curated pronunciation lexicon for a pre-trained G2P model or text-input. How reliable are implicit G2P models in generalisation compared to traditional G2P models trained on a lexicon? Do implicit G2P models reliably generalise to unseen word pronunciations? Is the extensive cost of maintaining a high-quality pronunciation lexicon in English really worthwhile? This chapter presents a theoretical analysis of implicit G2P modelling. Experiments are conducted with the DC-TTS system using text-input. This system belongs to the class of E2E-TTS that takes character input and predicts a mel spectrogram with a separate network for conversion to a waveform.

2.1 Motivation

S2S acoustic models in TTS such as Tacotron and Deep Voice 3 produce high quality speech without linguistic context features. Tacotron 2 for instance was shown to learn the pronunciation of words unseen in the training data (such as *supercalifragilisticexpialidocious*), pronounce tongue twisters (such as *she sells sea shells on the sea shore*) and pronounce the correct pair in common homographs (such as *read* in the present or past tense) [154]. However, whether the implicit pronunciation model

Datum	LJ	Nancy	VCTK
Total Word Types	14,750	18,695	5,839
Total Word Tokens	225,715	170,018	326,971
Total Sequences	13,100	12,095	44,070
Total Length (hours)	24	17	44
Mean Sequence Length (words)	17.2	14.1	7.4

Table 2.1: General Information on large TTS datasets

learnt from text-input is robust to modelling difficult word pronunciations remained unclear. If pronunciations can be implicitly modelled as suggested, a lexicon and the cost involved in its curation would not be necessary for TTS.

As mentioned in 1.3, previous work building TTS systems with graphemes observed lower intelligibility than when using phones in unit selection and with HMM acoustic models, even though more data could improve intelligibility. Furthermore, in ASR multiple works have suggested little difference in WER between text- and phone-input, especially when using S2S models with datasets of at least 300 hours. To what extent can word pronunciations be accurately learnt in these speech models? Is a lexicon still required in TTS for English or does an implicit G2P model learnt from a large set of text-audio pairs in training suffice?

2.2 Research Questions

1. How does implicit G2P modelling work in E2E-TTS?
2. How do implicit G2P models compare to G2P models trained on a pronunciation lexicon?
3. Is there a difference in DC-TTS when training on text- or phone-input?

2.3 E2E-TTS Datasets

2.3.1 Size

S2S acoustic models for TTS are typically trained with more hours of speech than previous SPSS acoustic models such as DNNs and HMMs (see [155]). The Linda Johnson (*LJ*) Speech corpus [156] and the Blizzard Challenge 2011 corpus (also known as

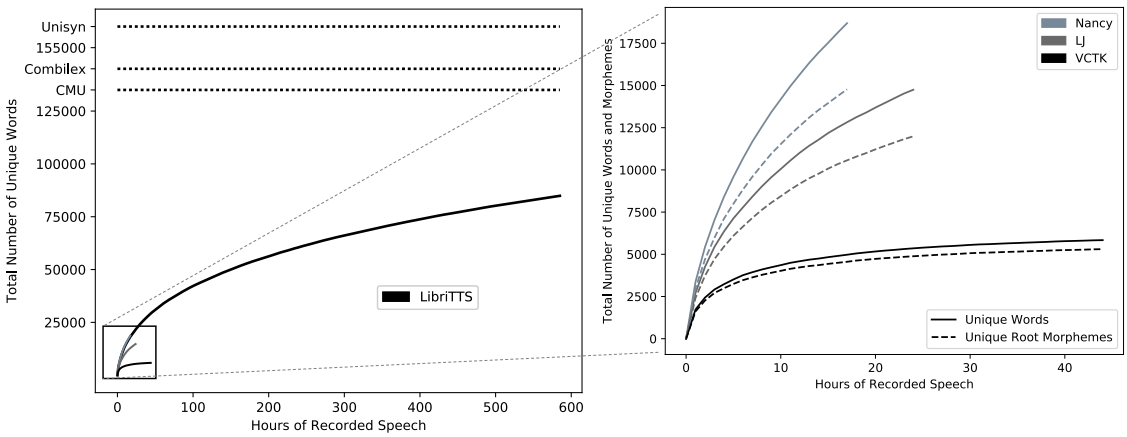

Figure 2.1: Cumulative coverage of unique word types per TTS dataset. The bottom left of the figure contains curves for LJ, Nancy and VCTK expanded in the graph to the right. The x-axis are hours of recorded speech in a dataset. The y-axis is the total of unique words. The curves expanded to the right also contain a count of unique root morphemes. The unique morphemes were initially counted to understand the relationship between cumulative root morphemes and word types. These follow the same zipfian pattern. Root morphemes were counted only for in vocabulary (IV) words to Combilex - the true number may be higher. The graphs show fewer unique word types than contained in common pronunciation lexica.

Nancy [157]) contain 24 and 17 hours of speech respectively. The open-source multi-speaker corpus from the Centre for Speech Technology Voice Cloning Toolkit [158] (*VCTK*) contains 44 hours of speech. Industry-only datasets at the time in Tacotron 2 and Deep Voice 3 contained 15-25 hours of speech. E2E-TTS has since been trained on more text-audio data. At the time of writing, the latest single-speaker work from the Tacotron team was trained on 39 hours [4]. The latest multi-speaker works were trained on 354 hours [135] and 243 hours [149] of audiobook data. Is a pronunciation lexicon still beneficial in TTS given these larger datasets?

2.3.2 Unique Word Coverage

To comprehend how much pronunciation knowledge was provided in these datasets, the unique word types and tokens of the types in the publicly-available datasets were counted. These numbers are shown in shown in Table 2.1. Usually, a data-driven G2P model is trained on word types that appear in a pronunciation lexicon: 135,000 types in the CMUdict, 165,000 in Unisyn and 145,000 in Combilex. For TTS in deployment

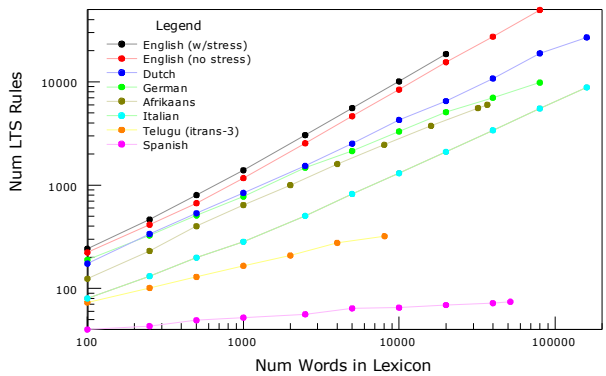

Figure 2.2: LTS (= G2P) rules in a decision tree as a function of lexicon size from [160]. The number of LTS rules in English increases with the number of words in a lexicon up to 100,000 words but common E2E-TTS datasets contain less than 20,000 word types. To what extent is pronunciation modelling weaker in English E2E-TTS than with the use of a pronunciation lexicon? This figure has been reproduced with permission from the original authors.

pronunciation lexica may be even larger as new pronunciations are entered according to need. However the E2E-TTS datasets presented in Figure 2.1 provide a narrower unique word coverage than pronunciation lexica.

For each sequence in the datasets, the new unique word types that appeared per hour of speech were counted and averaged. Initially, the comparatively smaller *LJ*, *Nancy*, and *VCTK* sets were analysed but later the calculation was applied to *LibriTTS* [159] (containing 585 hours) once the latter was released. The per-hour accumulation of unique word types is illustrated in Figure 2.1. The left of Figure 2.1, shows the curve for the *LibriTTS* dataset, and the right side is an expansion of the bottom left of the *LibriTTS* graph. The unique word types in pronunciation lexica are displayed above the *LibriTTS* curve, visualising the scale of the gap in unique word types.

G2P relations missing from the training data cannot be learnt. Such missing G2P relations could include place names and foreign words which would potentially be important for TTS in deployment. Implicit G2P models trained on E2E-TTS datasets may not be exposed to the same G2P relations as G2P models trained on pronunciation lexica, with presumably increased mispronunciations.

2.3.3 Zipfian Disitribution of Unique Word Types

The unique word types were observed to follow a zipfian distribution. Theoretically, one could accumulate more and more speech to achieve the same phonetic coverage

as a lexicon. However, obtaining comparable coverage from speech corpora would require an exponential increase in audio data, which would become increasingly costly and infeasible to process.

Contrast the information in Figure 2.1 with a plot from [160] shown in Figure 2.2. In Figure 2.2 the authors plotted the number of LTS (=G2P) rules in a decision tree as a function of lexicon size. In all languages[1], the number of LTS rules increased proportionally (at different rates depending on the language) with the number of words - in the case of English up to a 100,000 words. But the number of unique words in datasets used to train E2E-TTS models is typically below 20,000. Are E2E-TTS datasets with less than 20,000 unique words sufficient for errorless pronunciation modelling with S2S acoustic models?

2.4 Simulated G2P models

2.4.1 Method

In an attempt to quantify the performance of implicit pronunciation modelling in E2E-TTS, implicit G2P models were emulated by training explicit G2P models with TTS training data. A model was trained on a lexicon as a baseline. This analysis allowed for a quantitative comparison using the WER and phone error rate (PER), standard metrics in G2P modelling.

2.4.2 Lexicon

The training data were phonetised using the General American surface-form of Combilex. In this chapter, Combilex is used as the lexicon. In future chapters Unisyn is used instead. Combilex would usually require a commercial license whereas Unisyn is more accessible (free) for academic research. Both Combilex and Unisyn were preferred over the standard CMUdict due to the fewer word types and relatively low

[1]The experiments in this thesis focus on English but it would be interesting to conduct an analogous analysis in other languages. In Figure 2.2, there is an indication that English is not the only language for which LTS rules increase as words increase in the pronunciation lexicon (e.g. Dutch and German also exhibit similar rates whilst Spanish is notably flatter). Due to a lack of pronunciation lexica and publicly available datasets suitable for TTS voice building in other languages at the time this analysis focused on English. The release of the the CSS10 collection of single speaker datasets for building E2E-TTS voices in other languages [161] helps enable this analysis to be conducted on those languages too. The CSS10 resource was used to analyse pronunciation of an E2E-TTS models trained on French data in Chapter 3, but a large-scale multi-lingual analysis is presented as a key area for future work (see 6.3)

quality of phone entries in the latter (as recommended in [71]). Combilex and Unisyn derive surface form phones in multiple accents from a baseform metaphoneme lexicon [34] as explained in Chapter 1. In this experiment the General American (GAM) surface-form lexicon was used.

Ideally, a pronunciation lexicon would contain all the words contained in a TTS dataset. For instance, all words in the Arctic A and B script (commonly used to build unit-selection and SPSS voices) [162] were contained in Unisyn. However the larger TTS datasets contain out-of-vocabulary words.

Figure 2.3 presents a pseudo-Venn diagram of lexical content compared with Combilex. *LJ* and *Nancy* contained a substantial number of OOVs, indicated by the numbers outside of the Combilex circle. Since phone labels for these OOVs were unavailable they were not included when training the *LJ*, *Nancy* and *VCTK* models. Approximately 10% more data is used when training TTS models with *LJ* or *Nancy* than provided to the G2P models. While this may have disadvantaged the G2P models, the word types used to build the G2P models would still be fewer than contained in pronunciation lexica. All characters were lower-cased and all punctuation was removed (except hyphens denoting compound words and apostrophes denoting possessive *s*).

2.4.3 Input Types

The networks were trained with either:

1. isolated unique words (`Types`)
2. isolated word tokens (`Tokens`)
3. all words in a sequence (`Sequences`).

Although `Sequences` emulated learning in E2E-TTS in the category above, all models were tested on single words only. Testing on sequences could offer insights into how pronunciations are learnt contextually. For instance post-lexical pronunciations are an important aspect of implicit pronunciation modelling. However in this setup G2P models did not test post-lexical pronunciations. See Section 3.5 for an analysis of post-lexical pronunciations (in French).

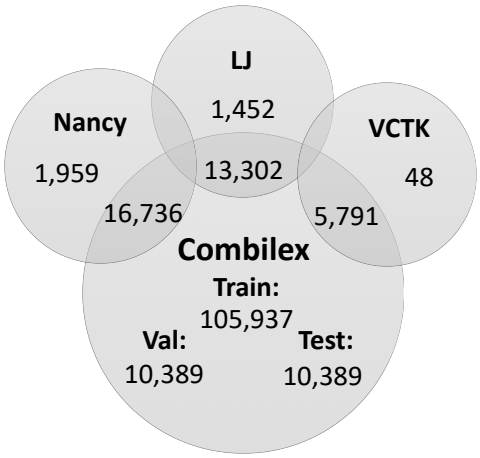

Figure 2.3: Words shared and not shared between Combilex and the E2E TTS training sets. NB: overlap also exists between *LJ*, *Nancy* and *VCTK*, but this was not shown to ease visualisation

2.4.4 Architecture

The OpenNMT [163] package in PyTorch, originally developed for NMT, was used to train Bidirectional Long Short Term Memory (BLSTM) models for the task of G2P. At the time, RNNs with LSTM units offered state-of-the-art results for G2P [82]. The relevant hyper-parameters were: 6 bi-directional encoder and decoder layers with 500 units each, a learning rate of 0.0001, dropout of 0.1, global attention [164], the ADAM optimiser and mini-batches of 64. The BLSTMs converged between 50,000 and 100,000 training steps.

The test set was a held-out set from Combilex also absent from the training datasets. The WER and PER were measured on the Combilex test set (containing 10,389 words) as shown in Figure 2.3. The WER was the Levenshtein distance at the word level between the predicted and correct strings, divided by the total number of words in the test set. The PER used the Levenshtein distance at the phone level. Both rates are expressed as percentages.

2.4.5 Results

Table 2.2 shows the results of the G2P models trained on TTS datasets. PER scores reflected WER scores. The baseline Combilex G2P model performed with PER = 1.1% and WER = 4.9%. The first two rows of Table 2.2 show the error rates for G2P models trained on single word types. The middle two rows show the error rates when trained on word tokens. The bottom two rows show the error rates for the models trained on

Input Type	Metric	LJ	Nancy	VCTK
Types	**WER**	52.9%	44.4%	82.5%
	PER	13.6%	10.3%	30.3%
Tokens	**WER**	64.0%	60.1%	89.5%
	PER	19.7%	17.7%	38.7%
Sequences	**WER**	42.5%	37.9%	57.7%
	PER	10.7%	8.9%	14.9%

Table 2.2: WER and PER for G2P models measuring implicit word pronunciation modelling from E2E-TTS datasets. `Types` are G2P models trained on unique word types. `Tokens` were trained with single words where the same word could appear multiple times during training. `Sequences` contained word sequences. All models were tested on unique word types. The Combilex baseline performed with PER = 1.1% and WER = 4.9%.

sequences (N.B. all sequences contained IV words).

For each input type the error rates were higher than the Combilex baseline. The error rates were particularly high for VCTK which had less than 6,000 word types. Models trained on the Nancy corpus, (with the highest number of unique word types), performed with the lowest error rates of the three datasets.

2.4.5.1 `Types` and `Tokens`

The error rates for tokens were higher than for `types` for all datasets. The higher error rates when training on `tokens` suggests a bias for the G2P relations contained in frequent word types was introduced. Since LSTMs (like other neural networks) contain latent, uninterpretable weights, such a bias is difficult to confirm. When analysing errors, I noticed groups of graphemes such as *th* in *pothole* were predicted as [D] (in X-SAMPA or [ð] in IPA) as pronounced in frequent words like *the*, *they*, or *there*. I reasoned that fewer unique word types and a frequent word token bias would degrade implicit G2P model quality in E2E-TTS.

2.4.5.2 `Sequences`

The error rates for `Sequences` models were the lowest of the input types. This suggested that longer sequences led to better results in the G2P model. To investigate how sequence length affected G2P performance, further G2P models were trained on the

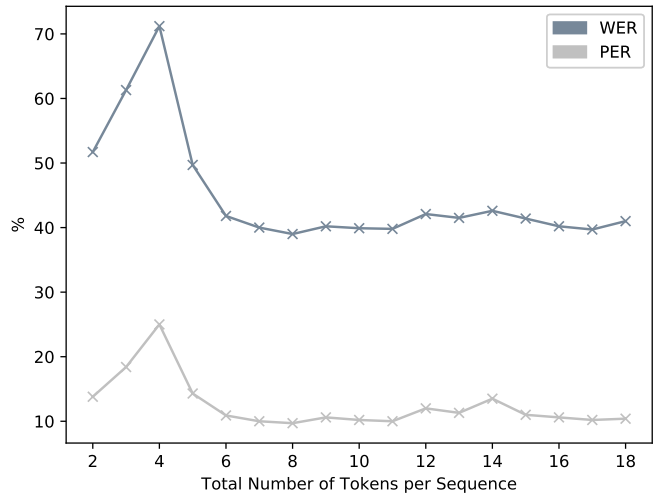

Figure 2.4: G2P error rates with increasing number of word tokens per sequence during training

Nancy dataset with a differing number of tokens per sequence.

The tokens were initially combined in their original order but randomising the tokens in the sequences made little difference, suggesting implicit word-level language model information had little effect on G2P performance. These results are presented in Figure 2.4, with highly variable WER for sequences of 2, 3, 4 and 5 tokens before levelling out for sequences of more tokens. For this particular G2P modelling setup (LSTM, *Nancy*), the results suggested input and output sequences of at least 60 characters, or 6 words, were an optimal length for training. E2E-TTS models are typically trained on sequences with an average of more than 7 words (see Table 2.1).

The above results suggest implicit pronunciation modelling is weaker than a G2P model trained on a lexicon. In the next section, there is a discussion of some limitations in emulating implicit pronunciation models in E2E-TTS using explicit G2P models. Further tests are then conducted to discover whether the G2P model results are paralleled in a DC-TTS model trained on text-input.

2.4.6 Constraints of Simulated G2P models

What is an appropriate way to show differences in pronunciation modelling when using minimally processed text input versus phone input in TTS? In the previous section implicit pronunciation modelling in E2E-TTS models was simulated by training explicit G2P models with E2E-TTS training data. However, this analysis was limited.

First, the implicit G2P models did not learn the same kind of implicit information as would E2E-TTS models. As already mentioned, some OOV word tokens were excluded from the models and post-lexical pronunciations were not included. E2E-TTS learns an acoustic representation rather than phone strings and OOVs may be implicitly modelled without the requirement for a corresponding phonestring like in G2P modelling. Importantly, in E2E-TTS with text-input, an intermediate phone sequence is not required. OOVs and post-lexical pronunciation are implicitly included in training without phones. Furthermore, the audio may contain pronunciation variation in speech style, rate and post-lexical pronunciation rules. Other components of E2E-TTS architecture may also influence pronunciation such as the type of encoder (RNN, CNN, transformer), the implicit duration method (e.g. attention mechanisms may skip over input sounds or produce babble noise) and waveform generation (which at the time may have used Griffin-Lim as in DC-TTS and Tacotron or employed a neural vocoder to convert mel spectrograms to waveforms such as in [113]).

Second, objective error rates do not reflect the plausibility of predicted sequences which are different from the true reference sequences. For instance, if the letter *i* in the word *tamil* were predicted with [*I*] (X-SAMPA)/[ɪ] (IPA) instead of [@]/[ə], this would not be a gross error. However, modifying the metrics to account for plausible alternatives, as attempted in [165], would not tell apart plausible from implausible predictions because the correct identification of alternatives would require hand-written rules. How E2E-TTS would pronounce these small variations was unclear. Given these constraints to the above analysis, how could emulated G2P models be shown to reflect implicit pronunciation modelling in E2E-TTS?

2.5 TTS of G2P Error Words

Initially, an E2E-TTS model was trained on *LJ* and a small listening test using some targeted stimuli was conducted. A larger-scale evaluation with targeted stimuli was conducted in Section 4.4. A selection of words were synthesised whose phonestrings were either a) correctly predicted or inaccurately predicted by the `LJ Tokens` G2P model in the previous section. The test words were embedded into the carrier sentence: '*Now we will say __ again.*'

A speech expert classified words in set a) as either *understandable/correct* or *incorrect/unrecognisable* and words in set b) as either *correct*, *wrong but recognisable* or *unrecognisable*. 100 correctly predicted words were randomly sampled, and a further

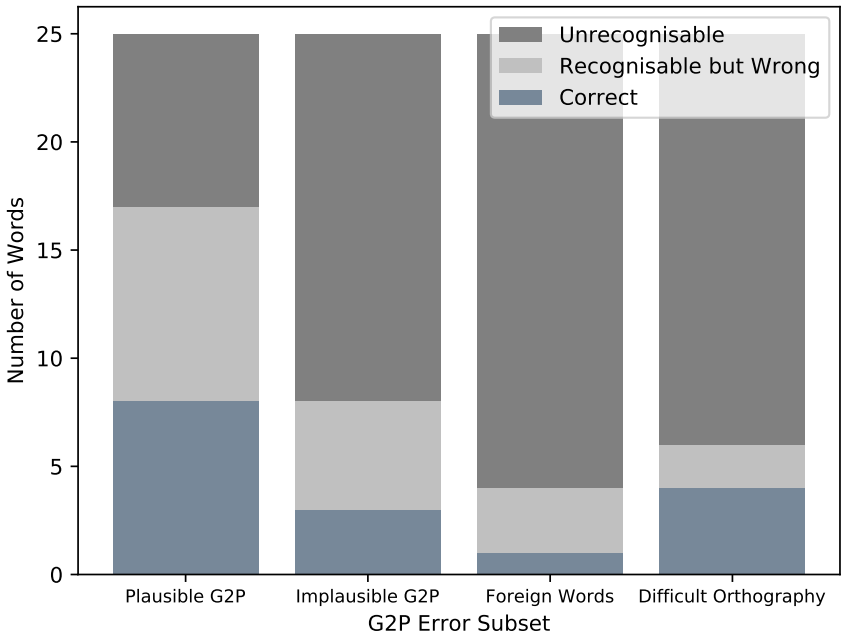

Figure 2.5: Expert listener judgement on E2E TTS for G2P error words.

100 incorrectly predicted words were hand-selected according to 4 categories:

1. where G2P gave plausible alternatives (*Plausible G2P* e.g. *tamil*)
2. where G2P gave inappropriate alternatives (*Implausible G2P* e.g. *loophole* as [l u f ə ɫ])
3. foreign names or loan words (*Foreign Words*e.g. *Flaubert* and *karate*)
4. English names with difficult orthography (*Difficult Orthography* e.g. *Loughborough* and *Worcester*)

2.5.1 DC-TTS

An open-source implementation of Deep-Convolutional TTS was used (DC-TTS [112]) with normalised text as provided in the LJ dataset. In early 2019, this was an open-source E2E-TTS system that produced high quality speech and was used by others in the field at the time (e.g. [3], [166]).

DC-TTS uses convolutional text and audio encoders known collectively as a text2mel (T2M) network to consume the input and predict 'coarse-in-time' mel-spectrograms.

Words	E2E Pronunciation (IPA)
loophole	[l u f ə ɫ]
goatherd	[g a ð ə d]
gigabytes	[g ɪ g æ b ɪ t s]
anchoring	[ə ʃ ɔ ɾ ɪ ŋ]
Loughborough	[l əʊ b ə r əʊ]
McElroy	[s ə l r i]
Siobahn	[s i əʊ b æ n]
ASCII	[ə s aɪ]
karate	[k ə ɾ ə t]
maoist	[m əʊ ɪ s t]
Flaubert	[f l ɑ b ə t]
Eduardo	[ə d o ɾ d u]

Table 2.3: Phonetic Transcriptions (IPA) of DC-TTS with text-input. The pronunciations are incorrect.

Forcibly incremental attention was used and the Super Spectrogram Resolution Network (a CNN) was trained to refine the predicted course Mel spectrogram. The SSRN consumes mel-spectrograms and upsamples them in both time and frequency to produce a full magnitude spectrogram. Both networks were trained for 300 epochs. Finally, the Griffin-Lim algorithm was used to re-introduce phase to the magnitude spectrogram and thus create the output speech waveform. For more information about the architecture see [112]. From Chapter 3 onwards, an implementation of Tacotron was instead used for experiments with E2E-TTS models.

All LJ utterances were included in training, including those with Combilex OOVs as this was not a restriction with text-input. All waveforms were downsampled to 16kHz before training.

2.5.2 Results

Of the 100 words predicted correctly by the G2P model, 79 were understood by the speech expert and 21 were unrecognised. These words could be difficult to understand without context (e.g. *flutings* and *sluicing*). Slight mispronunciations could render words unrecognisable. The expert often mistook them for more common words, for instance *mesher* was misunderstood as *measure*. Despite these perceptual difficulties,

the overall trend was that the words predicted correctly by G2P models were also intelligible to the expert when synthesised by DC-TTS with text input.

G2P error words with plausible but formally incorrect G2P predictions were synthesised more intelligibly on the whole. This is shown by the *Plausible G2P* column in Figure 2.5, where the "Unrecognisable" bar is shorter than for the other categories. Specifically in this category, unrecognisable pronunciations produced by DC-TTS were caused by stress placement on incorrect syllables. For example, the first syllable was more prosodically salient than the second in *regina*. Stress was an attribute of pronunciations not emulated in the G2P model analysis above. In [87], G2P models were jointly trained to predict stress and syllabification with improvements found in multiple languages when including stress information. In Chapter 3, some improvements in stress are observed when subword decomposition is applied to text-input (morphemes in English and syllables in French). However, when attempting pronunciation correction when mixing text-input with other lingustic representations in Chapter 5, stress markers are found not to be beneficial.

The other 3 categories of inaccurate G2P were associated with larger proportions of unrecognisable E2E pronunciations. Samples of such words with their pronunciations are in Table 2.3. Audio samples are available online[2].

2.5.3 Discussion

2.5.3.1 Frequency of Pronunciation Errors

To gain a perspective on the scale of the pronunciation problem, the frequencies of error words from the `LJ Tokens` G2P model in the British National Corpus (BNC) were plotted using estimated frequency data from [167]. The brown squares each represent a count per million for individual words in the BNC. The green crosses demonstrate the frequency of the `LJ Tokens` G2P model error words in the BNC (from which the *Out-LJ* test set is derived in Chapter 4 onwards). For ease of visualisation less than 1 count per million words were rounded up to 1 and the top 30 words in the BNC (functional items like 'the') were exluded from the top frequencies. All words containing digits and special characters (@, &, % etc were also removed to aid visualisation).

Only 699 out of 6648 error words had counts higher than 1 per million and the most frequent error word had an estimated count of 1154 per million. Figure 2.6 shows the majority of error words would be rare in everyday texts. However, the

[2]Audio samples at: `http://homepages.inf.ed.ac.uk/s1649890/lts/`

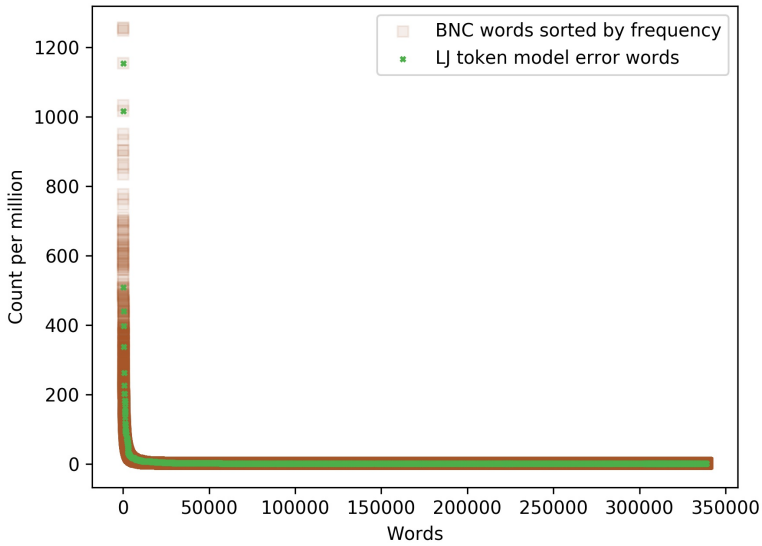

Figure 2.6: Counts of `LJ Tokens` G2P model error words in British National Corpus (BNC). The 30 most frequent BNC words were omitted for ease of visualisation. The figure demonstrates that the majority of G2P error words occur with a single count per million in the BNC. Nevertheless the ability to control pronunciation and correct errors (of names especially) is still important for TTS.

ability to control pronunciations is important in deployment and the problem should not be disregarded merely because the majority of pronunciation errors are for single count words. For instance, it would still be important for the names in Table 2.3 to be pronounced correctly for voice assistants (e.g. *Siobahn*).

2.5.3.2 From G2P to TTS

In the DC-TTS model trained with normalised text above, words with difficult G2P relations were mispronounced. There is pronunciation knowledge that cannot be learnt from text alone but that requires extra knowledge for disambiguation. The front-end composed of text normalisation, a lexicon and G2P model is used in front-end processing for this purpose. Similarly in E2E-ASR [17] it has still been noted that for particularly difficult words such as foreign words and proper names a pronunciation lexicon is still beneficial for correct recognition even if the general performance between text-based and phone-based systems is similar.

But how effective was this listening test at demonstrating that the simulated G2P models reflected implicit pronunciation modelling in E2E-TTS? It showed some clear

correspondences between the explicit G2P models trained on TTS datasets and the pronunciation of E2E-TTS output. However, the test was conducted on a small scale and was subjective: only 200 stimuli in total evaluated by 1 speech expert. In Section 4.4, the reliability of E2E-ASR for transcription of difficult G2P words is assessed. An objective and larger-scale intelligibility experiment is then conducted using transcriptions of 3,000 G2P error words from multiple systems in Sections 4.4.4.2 and 5.5.

The broader question for DC-TTS remained however: while some mispronunciations can be observed in targeted stimuli when using text-input, what about general speech quality between text- and phone-input? The experiments in the next section sought to answer this question with DC-TTS.

2.6 MUSHRA with Phone Label Corruption

2.6.1 Motivation

In the previous analysis, implicit pronunciation modelling of E2E-TTS with text-input was simulated by training explicit G2P models with E2E-TTS datasets. DC-TTS was also shown to mispronounce common error words from these G2P models.

G2P modelling in English is particularly difficult due to its notoriously irregular spelling. Graphemes in English are pronounced differently depending on the surrounding 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*. While work previously mentioned ([8]) shows phone contexts may be modelled (when using an input sequence of characters), an encoder with text-input would have to learn disambiguate these contexts which could introduce confusion into the G2P relations during training. Given the highly variable spelling in English, the pronunciations of some words (e.g. foreign words and proper names) are not easily disambiguated.

In this section, results are presented from a MUSHRA listening test where DC-TTS was trained with differing amounts of incorrect phone labels to find the approximate phone-label accuracy of using text-input.

The following analysis was conducted with my colleague Jason Fong. I created the training/test transcripts and Jason Fong trained all of the TTS models. We ran a listening test and published our results together.

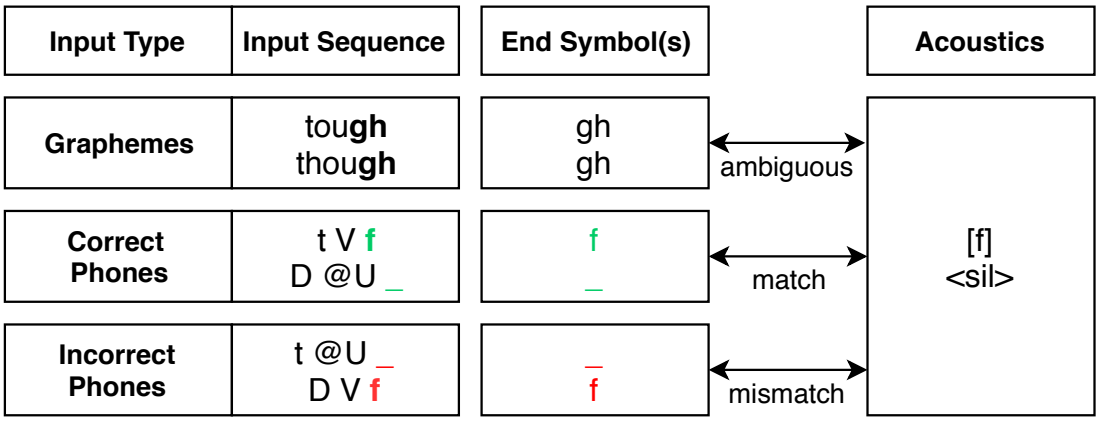

Figure 2.7: Simulation of corrupted phone input. The graphemes *gh* can have 2 pronunciations depending on a context learnable by the encoder. The correct phones contain no phonetic error and are gold-standard. Incorrect phones have mismatched pronunciations for *th* in *though* and *tough*. For each input sequence pair, the pronunciation should correspond to [f] and <sil> in the acoustics. The underscore symbol '_' represents a missing phone from the prediction, and '<sil>' represents a missing sound. Using correctly predicted phones results in a match between input 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 ambiguous, to what extent could graphemes have a negative effect on the training of the acoustic model?

2.6.2 Method

2.6.2.1 Creation of Training Transcripts

Figure 2.7 shows how text-input may be ambiguous in its G2P relations. The ambiguity in letters may be simulated by using incorrect phones. The letter cluster *gh* may represent either silence (<sil>) in *though* or a voiceless labio-dental fricative ([f]) in *tough*. A range of training transcripts with incorrect phone labels was generated to serve as a proxy for text-input with differing amounts of words of irregular pronunciation.

Phone transcripts were created using the following methodology. First, a 'gold standard' 0% WER transcript was generated via look-up using the full (Combilex GAM) lexicon. These phones were as accurate as possible given the large dataset size. Any sentences containing OOVs to Combilex were removed from the transcripts. Second, phonetic transcriptions were corrupted by varying the proportion of the train-

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 2.4: Description of phonetisation and WER of each training transcript. Input column denotes method used to phonetise transcript. Ratio column denotes % of full lexicon used, or the entries replaced by G2P predictions.

ing text phonetised by lexicon or by 2 different G2P models: a classification and regression tree (CART) model from Festival [168] and the BLSTM model described in Section 2.4.4 were used. In this way we compared text- to phone-input with varying phone-label accuracy.

Table 2.4 presents the breakdown of phonetisation in the transcripts. The *Ratio* column shows the percentage of phone labelling by G2P model. The *WER* column shows the WER of each training transcript. The error rates report the WER of the transcripts, not performance on a test set.

Recall some G2P errors may in fact be plausible variants, such as a prediction of the word *tamil* with [I] (in X-SAMPA) instead of a schwa phone [@]. The extent of plausible variants in the phone-transcripts was not measured as it would have required a manual review of all the errors.

2.6.2.2 Listening Test

The transcripts in Table 2.4 were input to 6 DC-TTS T2M networks. We chose a MUSHRA [169] test over a Mean Opinion Score test (MOS) to obtain a comparison between systems for each test stimulus [170]. We chose to measure naturalness as it is a recognised standard listening test metric in the field. We wanted a general idea of TTS quality when using text- and phone-input. In the test set, we only used words where the text had unambiguous G2P relations so they were easily predictable from graphemes. We selected 20 utterances from the wider set of 242 test utterances containing words

that did not require any disambiguation via the traditional front-end. Homographs[3] and abbreviations were excluded and any numbers were verbalised. Any possible errors resulting from grapheme ambiguity 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 to test all phone-based systems.

We recruited 30 English native speakers as listeners, paid £8 each for 45 minutes. The listening test was conducted in purpose-built listening booths. We included copy-synthesis of natural recordings as gold-standard upper bounds. We used the student t-test to measure the significance of each system's scores using the Holm-Bonferroni method for error correction.

2.6.3 Results

Figure 2.8 displays the results of the MUSHRA test. Participants were instructed to raise the score of the highest quality voice (natural) to 100, which is evident in the results. No such stipulations were made for other voices, and all systems scored below 52% on the naturalness scale, including the 0% WER transcript. 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[4].

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 intelligibility may have been affected. In subsequent chapters a different sequence model and a high quality neural vocoder are used without Griffin-Lim.

There was a 23.5% relative drop in the naturalness score when DC-TTS was trained on graphemes (`let`) rather than phones from the full lexicon (`100combi`), from 51.8 to 39.6. They were significantly different with $p < 0.0005$. In phonetic corruption terms, there was no significant difference in the naturalness scores of `let` and the phone-based systems trained with 25% WER (`100neur`).

The differences between the three best performing systems `100combi`, `50cart`, and `50neur` were 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, the proportion of the words with plausible variants was unclear.

[3]Predicting the correct pronunciation of homographs may require additional linguistic features [171]. These potentially pose another challenge to E2E-TTS with text-input (see Section 6.2.6)

[4]As may be heard in samples online: `https://jonojace.github.io/SSW19-comparison`

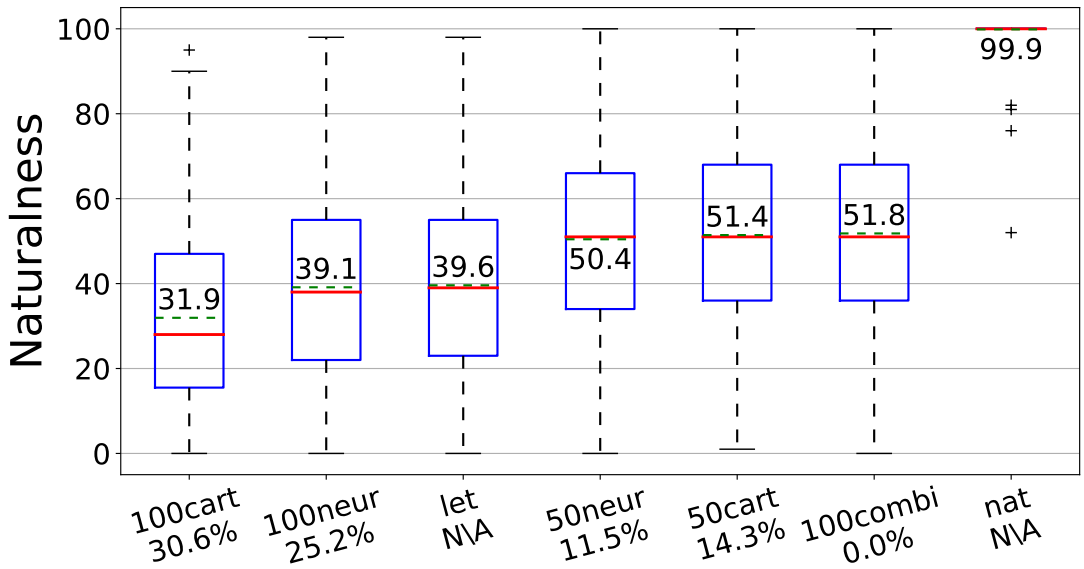

Figure 2.8: MUSHRA results. Solid red lines are medians, dashed green lines are means (also numerically labeled), blue boxes show the 25th and 75th percentiles, and whiskers show the range of the ratings, excluding outliers which are plotted with $+$. Percentages below system names indicate phone WER of their respective training transcript.

Incorrectly predicted phones could have been acoustically similar to the corresponding speech data and may not have greatly degraded the acoustic model. Unfortunately, teasing apart which predictions are viable and which are implausible (i.e those inaccruate phones that would have a detrimental effect in training) is non-trivial. Nevertheless, the naturalness scores of systems `let`, `100cart` and `100neur` were significantly lower than of system *100combi*.

2.6.4 Related Work

2.6.4.1 Grapheme- or Text-input is not a *Learned Text Encoding*

With the adoption of S2S models for SPSS, context features have been shown to be redundant by the use of a *learned text encoding* when using S2S models [3]. By *learned text encoding*, the authors of this paper mean the substitution of framewise context features for a text-encoder consuming character input to learn speech in context. The authors of [3] do not mean the use of text-input since all systems in their experiments use phone-input.

Work	Language	Dataset	Character Encoder	Vocoder	Sig. Improvements with Phones?
[111]	English	20 hours – single speaker (proprietary)	Deep Voice 3 (CNN)	Griffin-Lim	✓(mispronunciation test)
[174]	English	18 hours - single speaker (LJ subset)	DC-TTS (CNN)	Griffin-Lim	✓(MUSHRA)
[140]	English	385 hours – multi-speaker (proprietary)	Tacotron 2 (CNN + LSTM)	WaveRNN	✓(MOS)
	Spanish	97 hours – multi-speaker (proprietary)			✓(MOS)
	Mandarin	68 hours – multi-speaker (proprietary)			✓(MOS)
[173]	English	17 hours - single speaker (Nancy)	Tacotron 2 (CNN)	WaveNet	✓(MOS)
			Modified Tacotron (CBHL)		✗(MOS)
[4]	English	39 hours - single speaker (proprietary)	Tacotron (CBHG)	Griffin-Lim	✗(MOS)
				WaveRNN	✗(MOS)
				Flowcoder	✗(MOS)
			Wave-Tacotron (CBHG + LSTM)	n.a	✓(MOS)
[20]	English	260 hours – multi-speaker (proprietary)	EATS (CNN + RNN)	n.a	✓(MOS)

Table 2.5: A comparison of E2E-TTS with text- and phone-input. With the Deep Voice 3, DC-TTS and Tacotron 2 systems phones are significantly better, even in the Spanish multi-speaker model. In [173] it was proposed that the simplified CNN encoder adopted in Tacotron 2 [7] contained fewer parameters in the model and was less effective at disambiguating G2P relations in English. The non-significant finding with the CBHL encoder in [173] was repeated with the CBHG encoder using Griffin-Lim, WaveRNN and Flowcoder in [4]. However, when the CBHG was used in a fully E2E system in Wave-Tacotron there was a significant difference between text- and phone-input. The multi-speaker EATS model also found a significant improvement when using text- instead of phone-input. These results suggest a larger number of parameters may increase grapheme disambiguation during training. A systematic comparison across dataset size and languages with differing G2P complexity is suggested as future work.

2.6.4.2 Discussion Comparing Text- and Phone-input

In the above listening test, a significant difference was found between text- and gold-standard phone-input. The text-input system performed with naturalness similar to a training transcript with a WER of 25% when using DC-TTS with the Combilex lexicon, phoneset and G2P models. Subsequent works in S2S acoustic modelling for E2E-TTS have also compared text- and phone-input with different architectures. A summary of these works is provided in Table 2.5. The results on aggregate suggest that Tacotron 2 models that use CNN encoders consistently underperform when using text-input, as corroborated by results in [111], [172], [173].

However, there is evidence to suggest the ability to disambiguate character contexts in English spelling may depend on the encoder used. In English, the most systematic review conducted between architecture-type and character-input has been [173]. In this study, the convolution bank (CB), highway network [175] (H) and gated recurrent

unit [176] (G) components used in Tacotron (together: CBHG) were compared to the simplified CNN architecture of Tacotron 2. In their experiments G was replaced with an LSTM (CBH**L**). In a MOS test, a significant difference between graphemes and phones was observed when using the CNN encoder but not when using the CBHL encoder. The authors proposed that with more parameters, the CBHL had the capacity to better disambiguate difficult G2P relations in English during training which leads to naturalness scores akin to using phone-input[5]. A non-significant difference was also observed with a CBHG encoder in [4], when using Griffin-Lim, WaveRNN and Flowcoder for waveform generation. For the implicit task of disambiguating English spelling during training, these findings suggest that by increasing the number of parameters in the model, disambiguation via character-input is improved. This suggests E2E-TTS may have to consider a trade-off between reducing parameter size in a model and using grapheme or phone-input for overall quality.

Relatedly, successful learning from text-input may also depend on the method of waveform generation. When the authors of [4] tested their proposed Wave-Tacotron method using a normalizing flow decoder instead of a neural vocoder, a significant difference between grapheme- and phone-input was observed. Also in the multi-speaker EATS model (which used a spectrogram discriminator in a generative Adversarial Network (GAN)), phone-input significantly outperformed grapheme-input.

The comparison between text- and phone-input could be broken down further. What effect does dataset size play on the performance of either text- or phone-input? What about the difference between single-speaker and multi-speaker models? The results of [20], [172] suggest that when using a large amount of data (in the hundreds of hours), phones are more helpful than multi-speaker models. But how accurate must the phone labels be in multi-speaker models to outperform text-input? Can a G2P model suffice or is a pronunciation lexicon needed for labelling? Are surface-form phones required or could phones from a mismatched lexicon be used? Would an approach based on metaphonemes be more efficient for multi-speaker E2E-TTS?

What about differing G2P complexity across language (as shown in Figure 2.2)? In [172], the multi-speaker model trained on Spanish data still found significant improvements with phones. This result is particularly interesting since Spanish has relatively less complex G2P rules than English. Why do phones help in Spanish?

These are important questions which matter in appraising the value of the pro-

[5] A similar observation was made in the Tacotron paper where the authors noted that the CBHG module made fewer mispronunciations than a multi-layer RNN [2]

nunciation lexicon in E2E-TTS. A systematic study of the interactions between text- or phone-input and the following design decisions would make for interesting future work: the encoder type (CNN, CBHG), dataset size (see [155]), single-/multi-speaker modelling, waveform generation (fully E2E or with separate neural vocoder), the *phoneme* type (surface-form phones or metaphonemes), the accuracy of phone labels (from a pronunciation lexicon matching speaker-accent or from a G2P model).

The only work in Table 2.5 to use a targeted test was [111]. The authors created a set of 100 sentences which represented text normalisation and pronunciation challenges for deployment (e.g. abbreviations like *FBI*). The authors found that when trained on text-input, there were 19 cases of character skipping and 35 mispronunciations as opposed to 3 cases of character skipping and 4 mispronunciations when mixing representations with grapheme and phone-input (see Chapter 5). This finding complements the mispronunciation results shown in Table 2.3 in DC-TTS. Mispronunciations and differences in stress were also noted as the primary cause of perceived differences with natural speech in Tacotron 2 which was trained on normalized text sequences. Using the same 100 sentences as in [111], 6 sentences contained mispronunciations and 23 exhibited incorrect prosody. The MOS/MUSHRA tests presented in Table 2.5 do not explicitly evaluate the pronunciation of difficult words. The issue of general listening tests diluting the effects of potentially important differences between TTS systems was a point also made in [177]. Targeted stimuli are thus important to assess implicit pronunciation modelling of E2E-TTS. With targeted stimuli in [111], the use of phones under representation mixing explicitly improved pronunciations in output speech. The value of the pronunciation lexicon may more concretely be shown by using targeted stimuli.

The targeted stimuli in my own analysis were error words from a S2S G2P model. In the DC-TTS system with text-input, there were issues in implicit G2P generalisation around morpheme boundaries such as *loophole* and *goatherd*. In the former, the graphemes *ph* were pronounced [f], and in the latter *th* was pronounced [ð]. Could S2S G2P modelling be improved by using explicit morphological information? Could morphological information also improve E2E-TTS performance? These are questions tackled at the start of the next chapter.

It could be argued that given sufficient data a G2P model will learn to sufficiently generalise to unseen words such as *loophole* and *pothole* above. Indeed, via my own anecdotal observation, I notice words that contain ambiguous G2P relations can still be pronounced correctly. The issue lies in the reliability of G2P generalisation. Let us take

another example. Compare the pronunciation of *karate* with the word *rate* in English. If *karate* or other Japanese words with the subword *rate* are unseen in training, how is a G2P model to know from local character context alone how to pronounce *karate*? The correct pronunciation is not necessarily inferable from surrounding text. For certain words (foreign names and proper names make good examples), no matter how much data is used to train the model, the reliability that the pronunciation will be correct is less than when using phones. Therein lies the ultimate need for the pronunciation lexicon.

In Chapter 3, I also present work in French where pronunciations may not be easily predicted from local character context. *Liaison* is the insertion of sounds between word boundaries. In the traditional TTS front-end, *liaison* would fall under a post-lexical module. In the experiments, G2P error words and cases of disallowed *liaison* in French are tested.

2.7 Summary

2.7.1 Chapter Contributions

1. *How does implicit G2P modelling work in E2E-TTS?*

Since the G2P model in E2E-TTS is learnt implicitly in a joint encoder-decoder framework, E2E-TTS models are exposed to a narrower word coverage than traditional G2P models trained on pronunciation lexica. There is no established evaluation protocol to measure pronunciation modelling in E2E-TTS. In this chapter the implicit G2P models were simulated using explicit G2P models trained on the word coverage of common E2E-TTS datasets. This allowed for an evaluation of pronunciation modelling in E2E-TTS with WER and PER.

2. *How do implicit G2P models compare to G2P models trained on a pronunciation lexicon?*

G2P models trained on *LJ*, *Nancy* and *VCTK* performed with lower WER than when trained on the Combilex pronunciation lexicon. Error-prone foreign words and names in the simulated G2P models were also error prone when synthesised by the E2E-TTS model. G2P error words could serve as test cases for text-input E2E-TTS pronunciation models.

3. *Is there a difference in DC-TTS when training on text- or phone-input?*

Yes, a statistically significant decrease in naturalness was observed when using text-input. Subsequent studies have shown this finding does not generalise to all E2E-TTS models as elaborated in section 2.6.4.2, but importantly phone-input offers controllability over pronunciation.

2.7.2 Summary Remarks

In this chapter pronunciation modelling in E2E-TTS was analysed via a simulation with G2P models. The G2P models trained on E2E-TTS datasets scored higher error rates than a baseline G2P model trained on a lexicon. Stimuli containing G2P error words were synthesised to demonstrate similarities between explicit G2P models trained on E2E-TTS datasets, and DC-TTS trained with text-input. Mispronunciations for difficult G2P words were observed in DC-TTS trained with text-input.

Results from a MUSHRA comparison between text- and phone-input to DC-TTS were presented. While a significant difference between gold standard phone-input and text-input was observed, stimuli selected according to G2P error demonstrated pronunciation errors in DC-TTS with text-input. These results were placed in the context of subsequent work with newer architectures trained with more data. In light of further listening test results comparing text- and phone-input in E2E-TTS models, a large scale, systematic review of text- and phone-input across datasets and architectures was motivated (for the researcher(s) with the means to conduct such a large scale evaluation). However, implicit G2P modelling in E2E-TTS faces the challenge of reliably generalising to unseen character contexts where pronunciations may not be easily predictable from text as in foreign words and proper names.

In the next chapter, the role of the pronunciation lexicon is brought back into question with investigations on 2 areas where pronunciations may not be easily learnable from context. First, an investigation is conducted into the use of morphological boundaries in sequence-to-sequence G2P modelling, following the observation that separate morphemes in the pronunciation of words such as *pothole* and *goatherd* were not identified (as shown in Table 2.3). Second, an investigation is conducted into the pronunciation of difficult G2P words and the post-lexical phenomenon of *liaison* in French. Both investigations use an implementation of Tacotron that predicts mel spectrograms with a WaveRNN neural vocoder.

Chapter 3

TTS Experiments with Tacotron

In the previous chapter, it was argued that poor implicit G2P generalisation could cause mispronunciations in output speech of E2E-TTS systems with text-input. For TTS in deployment, reproducing adequate pronunciations for particular words may be very important. Following the observation from Chapter 2 that character clusters such as *th* and *ph* pose a generalisation problem for G2P in English, morphological boundaries are investigated here for S2S-G2P and E2E-TTS. In defence of the point that local G2P ambiguity in character contexts pose an inherent generalisation problem for pronunciation modelling with S2S models, results from preference tests in French are also presented. The E2E-TTS model in this chapter is an implementation of Tacotron that predicts mel spectrograms with a WaveRNN neural vocoder (see Section 3.3.5).

3.1 Motivation

In Chapter 2, G2P models trained on E2E-TTS datasets scored higher error rates than a G2P model trained on the Combilex GAM lexicon. G2P predictions of error-prone words were mirrored as mispronunciations in DC-TTS with text-input. In particular, mispronunciations at morphological boundaries (such as *th* in the word *pothole* or *goatherd*) were observed. It was also noted that learning correct stress could be problematic (e.g. *regina* and for longer words such as *sorrowfulness* or *incorrigibility*). Character clusters such as *th* and *ph* pose a problem for G2P in English, and pronunciation could be improved with decomposition of such words into their underlying morphemes or subword-units.

3.2 Research Questions

1. Does morphology improve G2P with S2S models?
2. What effects does morphology have on E2E-TTS audio?
3. Does morphology improve implicit pronunciation modelling in Tacotron?

3.3 Morphology for Subword Decomposition

Subword decomposition has shown improvements in sequence-to-sequence modelling for speech and language related tasks such as neural machine translation (NMT) [178], [179] text normalisation [49], [55], language modelling [180] and in ASR [18], [19], [181]–[183].

In NMT, subword decomposition improved the translation of rare or unseen source-target words, since components of such compound words were seen. For instance, in German *solar system* is one word *sonnensystem*, but the components *sonnen* and *system* occur as standalone words and as components in other compound words. When component words are delineated in sequence models, they become a *recurring subsequence* or *unit*[1].

Morphology should delineate meaningful sub-word units to resolve some pronunciation confusion in TTS arising from English spelling. For instance, *hanger* is composed of the root *hang* and bound morpheme *er*. These may attach to the root *coat* to derive *coathanger*. The characters *th* and *ng* are ambiguous in pronunciation: *th* could be confused for its pronunciation in *the* ([ð] in IPA), 'ng' could be confused for the pronunciation in *range* ([dʒ]). To a G2P model, the pronunciation of the sequences *th* and *ng* in *{coat}{hang}>er>* are clearer than in *coathanger*. Subword decomposition delineates phone contexts which should be unambiguous. Further examples of this delineation are shown in column *Format* in Table 3.1. As the *V* column in Table 1 shows, across the entire dataset the vocabulary is more than halved due to morphological boundaries.

By removing ambiguity from character contexts, delineating recurring subsequences is akin to using spaces to delineate words, except morphemes can appear in multiple words and thus be delineated into higher frequency subsequences. To further under-

[1]The use of the term *unit* should not be confused with the choice of grapheme or phone which may be interpreted as *units*. Here, it meant either *word* or *morpheme* units.

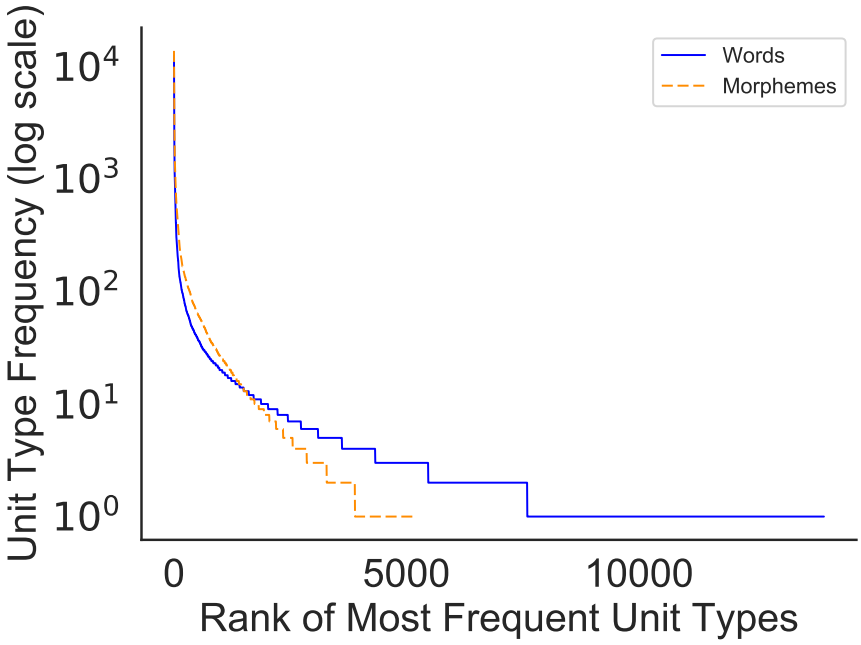

Figure 3.1: Total counts of most frequent units: words and morphemes in *LJ*. Splitting into morphemes reduces the vocabulary and increases the counts of seen units.

stand the proportion of words to morphemes for TTS in English, the number of recurring subsequences as words and morphemes were counted in *LJ*.

In Figure 3.1, the x-axis is the rank of the recurring subsequences (*Words* or *Morphemes*) - the rank runs in descending order, from left to right. The y-axis shows the frequency of the recurring subsequences in log-scale for ease of visualisation. The blue (*Words*) curve falls at higher frequency units more vertiginously than the orange (*Morphemes*) curve. It also has a long tail indicating a high total of recurring subsequences. The *Morphemes* curve stays above the *Word* curve as it kinks, indicating that the recurring subsequences of *Morphemes* occur with a higher frequency than of words. The *Morphemes* curve also has a shorter tail, demonstrating the set of recurring subsequences is fewer than the set of words.

3.3.1 Interpretability of Neural Models

Understanding the consequences of higher counts of smaller recurring subsequences on sequence-to-sequence learning is difficult. The weights of S2S models do not easily permit an explanation of explicit knowledge being modelled. Explaining implicit knowledge in neural network models is a complex task [184], [185]. Attempts to ex-

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

Table 3.1: 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. Note the input is still at the character level. In the TTS systems in this thesis, word boundaries are delineated using a word boundary token.

plain hidden units in TTS were covered in Chapter 1 in an attempt to observe phonetic context learning in E2E-TTS. Some other work, for instance in ASR has focused on recognising phone units in embedded representations of speech [186], [187]. Developing methods to improve the interpretability of neural models is also the focus of the BlackBoxNLP workshop [188], [189].

However, for the purposes here, in practice, multiple factors could influence the effect of using morphology such as the kind of subword decomposition (syllable, morpheme, unsupervised unit), dataset size, the language being modelled or the encoder type. When plotting training loss curves for the Tacotron models used, models that used morphology had more vertiginous drops in training loss than models without morphological information, as shown in Figure 3.2. But at the end of training, there were no large differences in training loss. Since these models were trained using a recipe in PyTorch without an explicit validation set, the validation loss curves for thse Tacotron models were unavailable.

An explanation of latent decision making when using morphology in neural models was not the aim of this experiment, but would be interesting future work. Following the method of [8], where a trained E2E-TTS model was treated as a classifier for phones, positional features and POS tags, contrasting inputs with graphemes, phones and morpheme boundaries could provide further insight. The main focus here however, is to analyse whether morphology improves G2P predictions and pronunciations in output audio.

3.3.2 Morphology in Unisyn

Though it does not feature in all lexica (e.g. CMUdict), morphological composition is indicated for all words contained in both Unisyn and Combilex. Unisyn provides entries interspersed directly with morpheme 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 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 of morphology could potentially add further benefit. For instance, canonical morphology [190] 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 3.1. Further information on the approach taken to morphology in Unisyn is provided in [191].

3.3.3 Supervised and Unsupervised Morphological Decomposition

The following experiments with morphology were conducted with Unisyn IV words. In practice, morphology would have to be predicted for OOVs. Supervised learning could be employed for morphological decomposition with S2S models. For example in [192], [193], the authors used an LSTM to segment text into morphemes. In [194] the author compared using uni- and bi-directional LSTM RNNs, CNNs with and without copy-attention and transformer models, bi-directional LSTM RNNs achieving the highest accuracy. Unsupervised morphological segmentation is also implemented in packages such as Morfessor [195]. Another common approach for subword decomposition is byte pair encodings (BPE - [178]). More recently, multiple methods for subword decomposition have been made available for researchers via SentencePiece [196]. In [197], BPE was compared to gold standard morphological boundaries in S2S-G2P models. While morphological boundaries led to significant improvements in WER, this was not the case with BPE. Whether subword units need to be linguistically symbolic is an area for future work (see Section 6.3).

Morphological decomposition (or segmentation) is separate from the task of morphological re-inflection. In reinflection, root words (e.g. *run*) are reinflected according to linguistic tags (e.g. the present participle *running*). Morphological re-inflection has

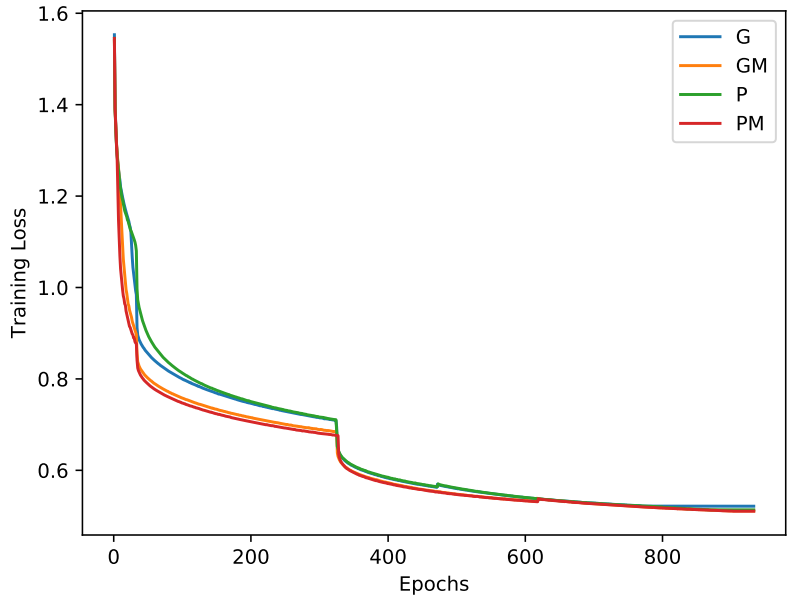

Figure 3.2: Training loss of G, GM, P and PM. In the first 400 epochs in the training schedule loss is reduced for GM and PM at a faster rate but by the end of training differences were small. An explicit validation set was not provided in the implementation.

been the subject of multiple *SIGMORPHON* challenges [198]–[201].

3.3.4 G2P Models

The effect of morphological boundaries was first evaluated in an explicit G2P model. The baseform lexicon of Unisyn was used. This lexicon was designed to be accent-independent and contained 160,000 entries. Prior to training the models, two partitions of the data were created as *random* and *disjoint* sets. For the *random* test set, 20% and 5% of Unisyn entries were randomly selected for the validation 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 appear 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· {hiccough} or any its derivations/inflections could not appear in the test set. In this way, the G2P

	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

Table 3.2: G2P error (%) with (GM) and without (G) morphemes

model's ability to generalise to unseen root morphemes was tested. The same G2P model architecture and training schedule was described in Section 2.4.4.

3.3.5 TTS Models

Out of the total 13,100 utterances in LJ speech (24 hours), 9871 utterances with IV items were used, totalling approximately 18 hours of speech. Utterances containing OOVs were left out 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 the intended purpose of understanding how, in principle, morphology effects E2E-TTS quality.

An implementation of Tacotron [202] was used. This implementation of Tacotron used a linear pre-net with dropout and a CBHG module to encode a series of one-hot input characters from a sequence into a single representation. The implementation also uses Location Sensitive Attention (LSA) from Tacotron 2 to reduce instability in output speech [7]. As noted in Chapter 1, S2S models with attention mechanisms sometimes babble, or fail to produce intelligible speech. Each Tacotron was trained for 350k, training steps, with a batch size of 32, learning rate of 0.001 following the default training schedule.

The implementation uses a WaveRNN vocoder based on [113]. A single vocoder on ground truth features was trained for use with the Tacotron models. A sampling rate of 16kHz was used.

3.3.5.1 MUSHRA Design

A MUSHRA listening test was conducted to test naturalness from 4 systems and a natural utterance as a hidden reference. 20 utterances were randomly selected from *LJ* that contained OOVs. The OOVs with correct morphology were added to the test text- and phone-input. As noted in Chapter 1, naturalness scores performed on a held-out

G input	GM input	G Pronunciation (Incorrect)	GM Pronunciation (Correct)
coathanger	{coat}{hang}>er>	[kəˈðeɪnʤə]	[ˈkoʊtˌhæŋɚ]
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]	[siŋkrəˈnaɪzəbl]

Table 3.3: 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.

test set do not test implicit pronunciation modelling directly. Targeted stimuli are used in Section 4.4. Nevertheless, given general performance increases when using subword information in other S2S tasks, a naturalness test was deemed worthwhile.

Example inputs to the 4 systems are shown in Table 3.1. The baseline was the grapheme-based (`G`) system, with base graphemes (text) only. The graphemes enhanced with morphology (`GM`) are interspersed with morphological boundaries. `P` and `PM` are the equivalent but using phones provided by Unisyn.

A closely-controlled listening test was held. The BeaqleJS platform was used [2] to implement a MUSHRA listening test [203]. The software 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. 30 native speakers were employed 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. Scores were aggregated identically to the MUSHRA in Chapter 2.

3.3.6 Results

The G2P results are presented in Table 3.2. On the *random* set, the WER was improved by 2% with the addition of morphological boundaries (from 9.9% to 7.9%). Moreover, the WER improved by 9.2% on the *disjoint* set containing unseen roots (from 32.6% to

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

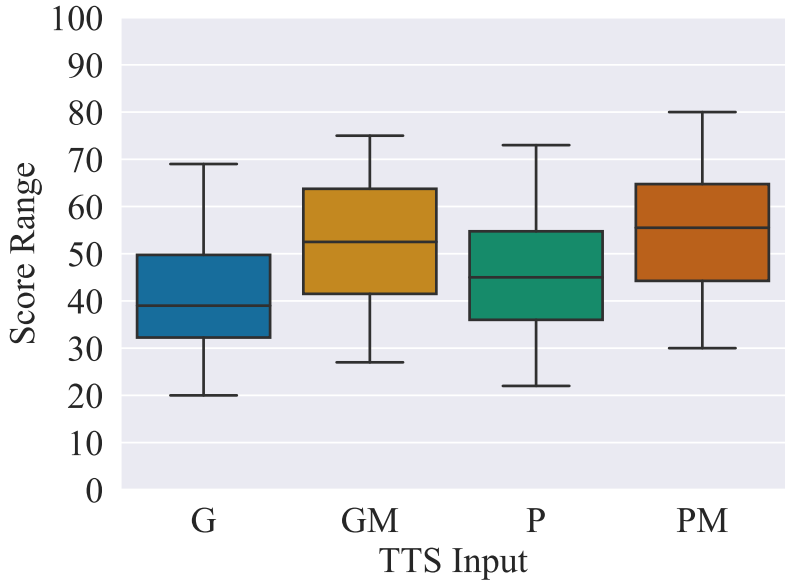

Figure 3.3: Range of scores from MUSHRA listening test of each system with phones or grapheme-based input with or without morphology

23.4%). These findings demonstrate that G2P modelling is improved when providing knowledge of morphological information on input. The margin was larger with a test set of words containing unseen root morphemes.

On a side-note, the WER for the *random* set appears comparatively low 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.

The average and spread of scores for each system in the MUSHRA is shown in Figure 3.3. Importantly, the MUSHRA is designed to demonstrate comparative results between systems. As such, the score for any one system should be interpreted relative to another, not as an absolute in itself. The hidden reference for this experiment was natural speech, and all the models exhibited unnatural intonation patterns due to a process of F0 averaging (see [134], [204] for attempts at improving the problem of F0 averaging with VAEs). The overall quality of speech samples was high – samples are available[3].

Supplementing both text- and phone-input with simple boundaries led 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

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

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]

Table 3.4: Improvements in TTS pronunciation from using phones: systems GM and PM

substantially improves neural TTS quality with this Tacotron implementation.

Error words from the `LJ Tokens` G2P model in the previous chapter were also synthesised. The system `GM` disambiguated pronunciation over grapheme clusters that system `G` pronounced incorrectly. Table 3.3 shows how adding morphology improves pronunciation of such words as *coathanger*, *upheld*, and *wobbliest*. A larger scale evaluation of G2P error words is conducted in Chapters 4 and 5.

System `P` system outperformed system `G`, but less so than the use of morphological boundaries. The improvement with `P` contrasts with the MOS test results of [4], [173] where similar Tacotron models reported no difference when using text- or phone-input. Direct comparison between these works is difficult however due to differences in datasets, quality of phone labels, encoder/vocoder architectures, and listening test (MOS/MUSHRA) types. This result further motivates a systematic review of input representations and as suggested in Section 2.6.4.2 but with the additional factor of subword decomposition. A template for such an analysis could be the analysis conducted in [18], where grapheme, phoneme and wordpiece units were systematically compared across the switchboard and LbiriSpeech datasets in E2E-ASR [18].

3.3.7 Targeted Stimuli

The above analysis of morphology contained a specific test with explicit G2P models and a naturalness test in Tacotron. The pronunciations of some anecdotal stimuli were also transcribed to demonstrate mispronunciations. To specifically measure implicit

pronunciation modelling in Tacotron, linguistically motivated stimuli such as words of inaccurate G2P may reveal more differences than naturalness scores from held-out sentences. To this end, a larger set of targeted stimuli is used in Section 4.4 for an intelligibility evaluation using ASR transcriptions.

3.3.8 Discussion of Morphological Input

In Table 2.5, it was noted that the learning from grapheme-input may depend on interactions between the encoder type and the waveform generation method (single-stage or with a separate neural vocoder). In light of subsequent work, further questions were asked about the interaction between the kind of data/ architecture and input representations used. In [18], [19] comparisons of grapheme- phoneme and unit types (characters v wordpieces) was conducted in ASR. An analogous study would also be worthwhile to quantify the value of subword decomposition in E2E-TTS: in particular the effectiveness of unsupervised methods for subword decomposition such as BPE or Morfessor (see Section 6.3).

Another interesting area of future work could be to adopt the approach taken in [8] to use a trained E2E-TTS model as a classifier to the tasks of G2P modelling or morphological decomposition to gain a more in-depth understanding of how much linguistic knowledge is implicitly learnt when text sequences are augmented with morphological boundaries.

3.3.9 Summary of Morphology Experiments

1. *Does morphology improve G2P with S2S models?*

Yes. G2P models performed with significantly lower WER and PER with morphological input (`GM`) on both random and disjoint test sets. Whether unsupervised forms of morphology (e.g. subword units) offer the same improvements would require further experimentation.

2. *What effects does morphology have on E2E-TTS audio?*

The `GM` Tacotron system performed with a statistically significant improvement in naturalness over system `G` and the speech sounded more fluent. In particular pronunciation errors over morpheme boundaries with the `G` model were corrected with the `GM` model. However, further experimentation would be required to discover how important gold-standard morphology is to this improvement over unsupervised methods.

3. *Does morphology improve implicit pronunciation modelling in Tacotron?*

   Yes, in this experimental setup (with gold-standard morphology). For deployment however several factors need consideration: an effective subword delineation or *grapheme-to-morpheme* predictor would be required. It is unknown whether similar benefits could be offered if the morpheme boundaries were predicted rather than gold-standard. This would be an interesting avenue for future work.

3.4 Other Languages

3.4.1 Text- or Phone-input?

To what extent is text-input a source of mispronunciation in other languages? As mentioned in Chapter 1, text normalisation is necessary in many languages for TTS in deployment (see [55]) . However, the extent of the normalisation may be related to the writing system in question.

In Eastern-Asian languages, text can be written with large sets of non-alphabetic characters. In Mandarin, a conversion from logographs to pinyin (a romanized, approximately phonetic alphabet) is necessary. This process is known as Grapheme-to-Phoneme (G2P) conversion but is applied to all input text for TTS, not only for OOVs as in English [205].

Interestingly in Mandarin, pinyin characters correspond to different tones depending on semantic context. The tones carry meaningful distinctions but are not easily predictable from the context alone. Therefore some semantic information is required beyond the phonetic level for correct pronunciation. This is known as the problem of polyphone disambiguation. Polyphone disambiguation is similar to homograph disambiguation in English[4] except polyphonic characters are commonplace in all text in Mandarin.

Manual rule-writing based on semantic context is very expensive and the number of interacting rules become difficult to scale in a similar way to G2P in English. While some works have proposed S2S models to generalise polyphone disambiguation [205]–[207], these still do not perform with 100% accuracy.

What of alphabetic languages other than English? G2P modelling in English is

[4]The pronunciation of *bass* depends on whether one is referring to a kind of fish or a musical instrument

particularly difficult due to irregular spelling, but it is notorious in this respect. In [5], no difference was found in using graphemes or phones when testing a Tacotron model in French. These findings are addressed in the next set of experiments in this chapter.

3.4.2 Subword Decomposition

What potential effects could input subword decomposition have on pronunciation modelling in other languages? Subword decomposition arguably benefits agglutinative languages (which exhibit extensive inflection and compounding of morphemes) more than English. Similar experiments to those conducted above in English were performed on Kiswahili data in [208].

In Kiswahili, text-based pronunciation ambiguities occur in the agglutination of foreign words such as English loan words. For example, in the phrase '*zinatake place*' (*they take place*), the English word *take* adopts the English pronunciation [teɪk], but usually the letter 'a' in Kiswahili is pronounced [ɑ] not [eɪ]. Furthermore, the letter 'e' is always pronounced in open syllables. In an error analysis, subword information improved the pronunciation of specific loan words. G2P WERs were improved by using morphological and syllabic boundaries. Furthermore, the morphological boundaries (from Morfessor) and syllabic information significantly improved naturalness over a text-input baseline.

As mentioned above, when training E2E-TTS for non-alphabetic languages, a conversion to phones is preferred to avoid large character sets. In Japanese, the hiragana and katakana syllabaries are phonetic and the logographic kanji can be converted to one of these alphabets. However, neither the writing system nor a phonetic sequence of characters denote pitch accents. Pitch accents are meaningful in Japanese and must be predicted from context [209]. This usually requires complex, rule-based text-analysis [210]. When implementing Tacotron for Japanese, contextual linguistic features such as the mora, syllable can improve naturalness, but not as much as pitch accent information [173]. The contribution of subword decomposition may also however depend on the particular evaluation set-up in each work, as in English. Future research could investigate the value of subword decomposition in Japanese and non-alphabetic languages further.

However, in other languages it may not only be a matter of correct phones- or subword decomposition. Other linguistic information which is derived manually is still required and presents a barrier to E2E-TTS without resources. It is not just English

with a problematic orthography. Modelling correct pronunciations from local character contexts may require linguistic information, as semantic information for polyphone disambiguation in Mandarin, or pitch accent information in Japanese. Below, another difficult-to-model-from-context pronunciation phenomenon is investigated: disallowed *liaison* in French.

3.5 Experiments in French

3.5.1 Motivation

As shown in Figure 1.4, the authors of [5] observed single graphemes in context can map to multiple phone sounds. Indeed, E2E-TTS models implicitly learn character contexts and correspondingly different pronunciations. The authors of [5] conducted a MUSHRA evaluation comparing text- and phone-input to a Tacotron implementation with a CBHG module. Listeners were also asked to rate the pronunciation of the samples on a scale from 1-5 in a MOS-style test. Text- and phone-input performed with no-significant differences in these tests. In addition, tongue twisters were included to test pronunciations also with no significant difference found. This suggests a pronunciation lexicon may add no benefit for E2E-TTS in French.

However, pronunciation phenomena are not easily generalisable from the local character context alone. The authors also noted that the system with either text- and phone-input produced errors in the pronunciation of *liaison*. *Liaison* is a process where linking sounds are inserted between words. Traditionally, it occurs during the "post lexical" module of the TTS front-end, after an initial phone string has been obtained from a lexicon lookup or G2P model. The plural possessive *mes* before a following consonant has no pronunciation corresponding to the *s* grapheme: *mes chats* - [me . ʃa]. But before a following vowel, the *s* grapheme corresponds to the pronunciation [z]: *mes amis* - [me. **z**a. mi]. The rules governing *liaison* operate at a deep linguistic level which are difficult to model. For instance, *liaison* cannot occur after a subject noun and a verb, e.g.: *mes amis arrivent* - [me. **z**a. mi ∅ a.ʁiv]. While data modelling of *liaison* has been tested with decision trees [211] and templates [212], the process is complicated further because its use is often stylistic and optional [213], consequently hand-written rules are often used for TTS.

The interesting question here is whether *liaison* produced by a Tacotron model is appropriate: i.e. avoided in particular disallowed linguistic contexts (*liaison inter-*

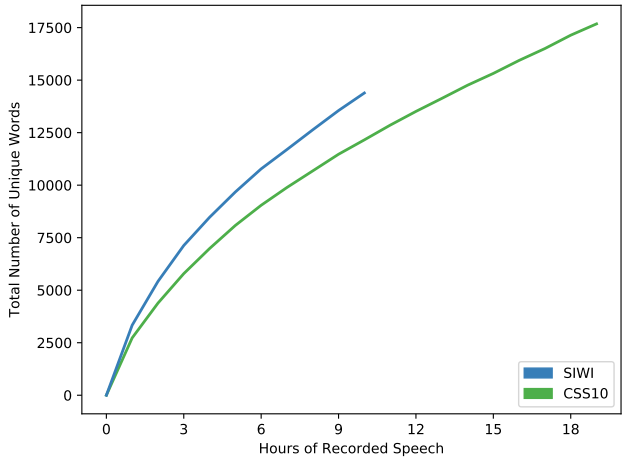

Figure 3.4: Total unique words in SIWI and CSS10 French TTS Datasets. The datasets cover fewer unique words than the lexicon used by MaryTTS which contained 112,130 unique word types. Unusual G2P relations not covered in the training data may not be predicted accurately, such as for foreign names.

dite). The French language has a highly active normative body called the Academy (*l'Académie Française*) who maintain a strict standard form of the language prohibiting insertion of *liaison* sounds in certain contexts, such as before the aspirated-*h* in combinations like *les **h**aricots* or *les **h**érissons* (see Figure 3.6). If sounds are inserted in these contexts it is not considered *le bon usage* (proper usage) of the French language. While speakers do not strictly obey all rules, the ability to control the pronunciation in such a context could be important to certain users/ in deployed use cases. Since *liaison* is not an easily predictable phenomenon from local character sequences, it provides targeted stimuli for a comparison of text- and phone-input to a Tacotron model. Since the use of *liaison* is optional in many contexts, the listening test focuses on cases of *disallowed liaison*. These are cases where sounds should not be inserted.

Another model was also trained to test the effectiveness of syllable boundaries on the quality of the Tacotron with phone-input in French. Following the benefits of morphology above in English and the use of syllables in Kiswahili in [208], Syllables were chosen for another interesting pronunciation phenomenon in French: *enchaînement*. *Enchaînement* occurs when the final sound of one word transfers to the first syllable of the next word. For instance, in *mon cher ami* the final rhotic of the word 'cher' is the onset to the syllable of the next word *ami* - [mõ . ʃɛ . ʁa . mi] with a consequent difference in stress. Could *enchaînement* be improved with syllable boundaries?

3.5.2 Research Questions

1. Does Tacotron in French with text-input mispronounce words with challenging G2P relations?

2. Does Tacotron in French with text-input learn cases of disallowed *liaison*?

3. Do syllable boundaries improve the pronunciation of *enchaînement* in French?

3.5.3 French Resources

To answer the above questions, preference tests with targeted stimuli in Tacotron models were conducted. A TTS front-end in French was required that would provide accurate phone strings during training and testing. In [5], eSpeak was used. This package generates phonestrings via rule-based G2P modelling without *liaison* post-lexical rules. Another front-end for French was available in MaryTTS, which outputs linguistic metadata such as syllables, and Part of Speech (POS) tags. The lexicon was based on the database Lexique [214], each word wherein was phonetized and syllabified using *LIA PHON* [215] whose PER is 1.3% (the syllable error is unknown). The default French voices in MaryTTS do not provide post-lexical rule-based phonetization such as *liaison*. *Liaison* post-lexical rules were therefore manually based upon the guide available in [216]. POS tagging was a core input attribute for the liaison module so the French MaryTTS front-end was modified to use the Stanford POS tagger [217] to ensure as high accuracy as possible[5]. The front-end of MaryTTS is designed on an XML-based Document Object Model and details on updating modules in MaryTTS pipelines is provided in [218]. For the phone-input systems below, the French front-end from MaryTTS was used, with its default lexicon and G2P model enhanced with *liaison* post-lexical rules.

19 hours of audiobook data recorded by Gilles G. Le Blanc were used for training, distributed as part of the open source CSS10 dataset [161]. 5% of the data from which test stimuli were randomly sampled were held-out for the CSS10 AB listening test. As in Chapter 2, the numuber of unique word types per hour in the CSS10 and SIWI datasets were counted and averaged. The CSS10 dataset was chosen for its larger and wider coverage of unique word types. The lexicon used by MaryTTS contained 112,130 unique word types. The distribution of word types is shown in Figure 3.4. The

[5]The maintainer of MaryTTS (Sébastian le Maguer) made the modifications to the POS tagger as described.

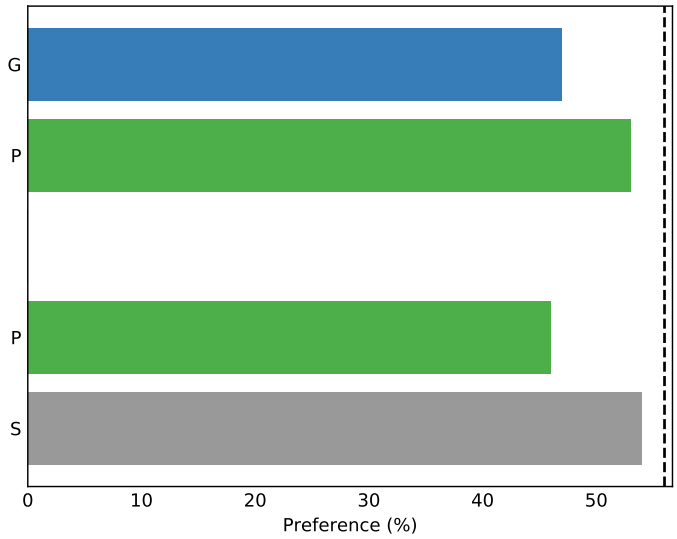

Figure 3.5: Results from preference tests using CSS10 stimuli. No significant differences were observed between grapheme-input (G), phone-input (P) and phones enriched with syllable boundaries (S). The significance level at p = 0.05 is shown by the black dotted line at x=57.

proportion of words with irregular G2P relations such as foreign names in the lexicon and datasets is unknown.

3.5.4 Listening Test Design

The AB preference tests were run on 10 sentences held-out from the CSS10 dataset between:

1. grapheme- (G) and phone- (P) input;
2. phone- (P) and phone-input enriched with syllable boundaries (S)

To test pronunciation with targeted stimuli, a G2P model was trained using the CSS10 data to identify words with difficult G2P relations. These words were then synthesised by the Tacotron model. The OpenNMT architecture was used as previously except with the lexicon from MaryTTS. 10 error words from the test set were placed in the carrier sentence: "*Il a dit ... encore une fois*" = "*He said ... again.*".

To test *liaison*, 10 sentences, each containing impossible or disallowed *liaisons* were synthesised. As noted in [5], impossible cases of *liaison* are problematic for

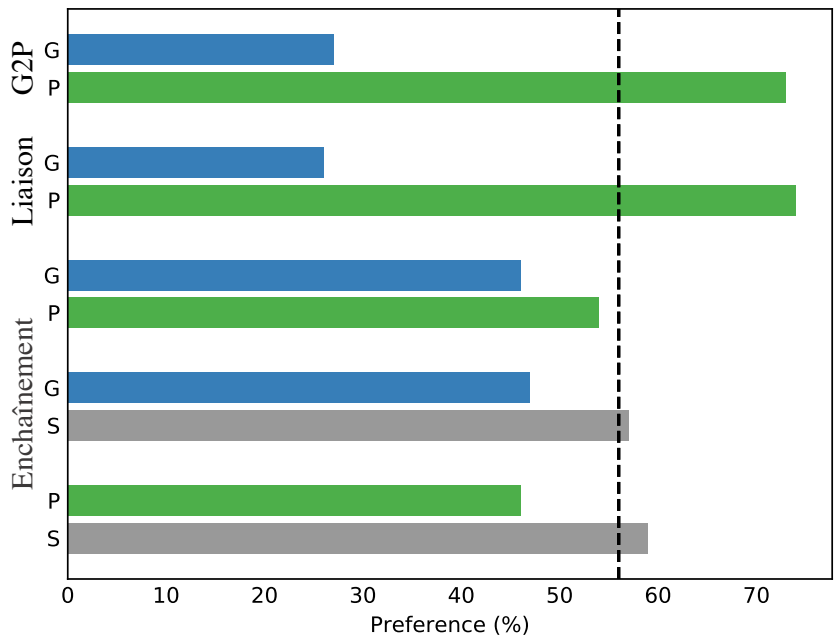

Figure 3.6: Results from targeted preference Test. The first tier shows G2P results, the second tier shows *liaison*. The last 3 tiers show results from the test with *enchaînement* stimuli.

Tacotron - for example where an *s* can be inserted before an aspirated-*h* as in *les_haricots*. The test compared output from the `G` and `P` models.

To test *enchaînement*, 10 sentences were created, each containing cases where the word-final consonant becomes the onset of the following word-initial syllable. The samples were compared from the `P` and `S` models.

AB preference tests were built in Qualtrics. Due to social distancing policies, an online listening test was held using the Prolific platform. In-person tests have the advantage of controlling listening conditions but there is evidence that online tests can lead to consistent results. For example, there were high correlations across 5 different sets of listeners across 5 different days in [219]. However, crowd-sourcing of TTS evaluations can be affected by noise, such as street noise and background TV-noise [220]. For control, participants were only allowed to take the test on a desktop and not a mobile phone. 30 participants took part. Participants were paid £5 per 30 minutes of their time. Participants were native French speakers and had no known hearing difficulties. For the general and targeted preference tests the accompanying question on each screen was: *Which clip has better pronunciation?/ (Quel clip a la meilleure prononciation?)*[6].

[6]Samples are available at: `http://homepages.inf.ed.ac.uk/s1649890/fren/`

Word	G (Incorrect)	P (Correct)
Miguel de Cervantès	[**d**igɛl də sɛʁvã**tz**]	[migɛl də sɛʁvãtɛz]
Les Coopers	[**te sko**pə]	[le kypɛ]
Monica Lewinsky	[**pw**anika lew**ẽs**i]	[monika lywinski]
Rio de Janeiro	[**tʁ**io də ʒanero]	[ʁio də ʒanero]
McLaren	[**klaʁno**]	[məklaʁɛn]

Table 3.5: IPA transcriptions of words of inaccurate G2P included in preference test. Mispronunciation of names by the G model are highlighted in bold. The pronunciations contained unusual sounds with some skipping for unusual character contexts such as *Lew**insky***. How are character contexts unseen in training data handled implicitly by G2P models?

3.5.5 Results

3.5.5.1 CSS10 Stimuli

The results from the general AB listening test are shown in Figure 3.5. No significant differences were found between the G and P systems, nor between the P and S systems. System S had been expected to perform with a significant preference over system P since the distribution of recurring subsequences when using syllables was lower than using words. However, as mentioned in Section 3.3.1, a comprehensive explanation of latent decision making is difficult. However there was an observed preference for S when testing the pronunciation of *enchaînement*.

3.5.5.2 G2P Error Words

The results from the targeted AB listening test are shown in Figure 3.6. The phone-input models had accurate phone labels for this targeted preference test. Listeners significantly preferred P over G in the G2P preference test. Some incorrect pronunciations by system G are shown in Figure 3.5. Listening to samples, there were mispronunciations for unusual grapheme sequences in French such as ***coo**per*, *rio*. Sounds were also skipped such as the 'k' in *Lewins**k**y*. There was also an inexplicable [o] sound inserted in *McLaren*. Increased skippings and mispronunciations were found with the targeted test of difficult words in [111] in English.

The mispronunciations are further evidence that gold standard phone-input can be

Input	Labels
G	Le**s** **h**aricots pousseront plus efficacement en plein air. Il a mi**s** **u**ne chemise.
P	[le aʁiko pusəʁõ plys efikasəma˜ ã plẽn ɛʁ] [il a mi yn ʃəmiz]

Table 3.6: *Liaison* inserts sounds at word boundaries according to complex rules, but inadequate insertion such as following an aspirated-h or between a past participle and a determiner was dis-preferred. Inadequate *liaisons* are highlighted in bold.

better for words with difficult G2P relations (earlier examples have been provided in Tables 2.3 and 3.4). A pronunciation is helpful for words with difficult G2P relations as presented.

Given the poorer quality of these 10 G2P error words for system G, it is surprising that G was given some preference by some listeners for some sentences. Indeed, it was difficult to control the listening conditions of listeners since the test was conducted via a crowd-sourcing platform.

3.5.5.3 Liaison Stimuli

Listeners significantly preferred P over G. In each case, G inserted liaison sounds where they should not have been inserted. It is interesting to note that some listeners did not mark G down. Speakers do not strictly obey all rules of *le bon usage* in French and *liaison* is often mis-used. It is difficult to know whether G was sometimes instead preferred due to differences in listening conditions (or other uncontrollable factors). Nevertheless, the scores demonstrate that system P (which did not pronounce *liaison*) was still preferred overall.)

3.5.5.4 Enchaînement Stimuli

No significant differences were observed, but there was a preference for system S over system P. With syllable boundaries replacing word-boundaries, prosodic breaks occurred between syllables and less so at word boundaries. Some examples of the differences in the breakdown are shown in Table 3.7.

Input	Labels
G	Le <> ciel <> est <> bleu <> et <> la <> mer <> aussi Les <> sept <> enfants <> ont <> raconté <> une <> histoire <> amusante
P	lə <> sjɛl <> ɛ <> blø <> e <> la <> mɛʁ <> osi le <> sɛt <> ãfa˜ <> õ <> ʁakõte <> yn <> istwaʁ <> amyzãt
S	lə . sj**ɛ . l**ɛ . blø . e . la . m**ɛ . ʁ**o . si le . s**ɛ . t**ã . fã . õ . ʁa . kõ . te . **y . n**i . stw**a . ʁ**a . my . zãt

Table 3.7: Input string differences with syllable boundaries. '<>' denote word boundaries, '.' denote syllable boundaries. The boundaries in the S system cross the word boundaries between 'ciel-est', 'mer-aussi', 'sept-enfants' and 'histoire-amusante'.

3.6 Discussion on Pronunciation Evaluation

So far, pronunciation errors have been shown when using text-input in the E2E-TTS models DC-TTS and Tacotron. In particular, cases where pronunciations are not easily learnt from local character context have been analysed. The listening test in French sought to demonstrate that by using stimuli targeted to aspects of a language in question, differences could be observed between text- or phone-input not observable in other kinds of listening tests (such as MUSHRA or MOS).

However, the targeted stimuli in these experiments have been small in number (in terms of stimuli and listeners) and subjective (online/in-person). Whilst an objective evaluation was presented with simulated G2P models in Chapter 2, insights from G2P modelling are limited. The most convincing evidence of mispronunciations in E2E-TTS have been anecdotally observed (e.g. placenames and foreign words in Tables 2.3 and 3.3). Ideally, evaluations could be conducted on diverse kinds of stimuli without the cost of running large scale listening tests. With the introduction of social distancing in early 2020, conducting evaluations without in-person tests became important. The next chapter assesses the reliability of ASR in measuring intelligibility of TTS output.

In particular, an E2E-ASR model based on a Transformer from ESPnet [153] is employed that does not use a pronunciation lexicon. Since E2E-ASR models have been observed to make errors for difficult G2P words in a similar way to E2E-TTS with text-input (e.g. see [17]), its reliability for exactly these kinds of words is also analysed.

3.7 Summary

3.7.1 Summary of French Experiments

1. *Does Tacotron in French with text-input mispronounce words with challenging G2P relations?*

Yes, foreign words were mispronounced with system G. It is important to note that phone-based systems may also mispronounce such words if phones are predicted via a G2P model instead of being hand-labelled.

2. *Does Tacotron in French with text-input learn cases of disallowed liaison?*

No, cases of disallowed liaison were mispronounced. The rules governing liaison are linguistically complex and Tacotron learns a generalised pattern rather than the underlying rules.

3. *Do syllable boundaries improve the pronunciation of enchaînement in French?*

In this listening test there was a small preference but further work is needed to establish benefits (if any) of syllable boundaries.

3.7.2 Summary Remarks

This chapter presented experiments using text-, phone- and subword unit- inputs to a Tacotron model. Morphological boundaries in English were found to improve TTS quality. Preferences were observed for phone-input in samples of G2P error words and disallowed *liaison* in French. Targeted stimuli revealed differences between systems that were not observed with stimuli from regular TTS test sets.

The need for a reliable objective metric to efficiently conduct large-scale evaluations was also mentioned. To this end, in the next chapter an analysis of ASR as an objective intelligibility metric for TTS (and in particular for pronunciation evaluation) is conducted.

Chapter 4

TTS Evaluation using ASR

This chapter investigates the use of automatic speech recognition (ASR) to evaluate intelligibility in TTS. The adoption of this approach to measure TTS performance (e.g. as in [4]) is a demonstration of the improvements in ASR in recent years. Whereas works (e.g. [21]) typically make use of a free ASR API, the analyses in this chapter make use of an open sourced Transformer-based model available from ESPnet [153]. This model does not use a pronunciation lexicon, which allows for an interesting comparison in how E2E-ASR and E2E-TTS deal with words of difficult G2P relations simultaneously. It could be argued that transcription of difficult G2P relations by an E2E-ASR system is not a sensible method of evaluation. But if there are no issues in pronunciation modelling by E2E-ASR and E2E-TTS systems, then there should be no problems.

4.1 Motivation

In this thesis, evaluations of pronunciations with targeted stimuli have so far been on a small scale and subjective. G2P as an objective metric was used in Chapters 2 and 3, but as explained, there were differences between simulated G2P models and E2E-TTS audio output. ASR could be used to transcribe E2E-TTS output audio, thereby offering an intelligibility-like metric.

Recently, authors have analysed TTS performance using ASR [4]. However, to my knowledge the reliability rankings according to intelligibility had not been investigated. Furthermore, insights from the evaluation of more stimuli than feasible in human listening tests had not been analysed in detail.

4.2 Research Questions

1. How does ASR compare to paid listeners when transcribing synthetic speech in the Blizzard Challenge?
2. How do confidence intervals over ASR WERs change as the number of TTS test stimuli is increased?
3. Do ASR transcriptions identify the same significant differences between system pairs as the paid listeners?
4. Are there any benefits to increasing the number of stimuli for ASR transcription?

4.3 Blizzard Challenge Re-evaluation

4.3.1 Objective Metrics

The development of objective evaluation metrics is crucial to the field of text-to-speech synthesis (TTS). Traditional listening tests conducted under controlled conditions are expensive, and the data collected may require extensive quality control [221], [222]. The drive for simpler and less expensive means for evaluation have resulted in use of metrics such as PESQ [223], MCD [224] and ViSQOL [225]. Recent work has also focused on the prediction of MOS for TTS [226]–[228] and voice conversion [229], [230] systems using neural networks.

However, an intelligibility metric measurable by WER or CER could potentially capture details of mispronunciations in TTS systems not captured in naturalness metrics (see Section 4.4.4). Previous work on objective intelligibility measurement has focused on speech in noise to evaluate speech enhancement algorithms [231]. This was the subject of the Hurricane Challenge [232]. Recent progress in ASR has enabled the use of ASR transcription as a more interpretable metric for intelligibility. A phone-based ASR system outperformed other objective intelligibility measures for evaluating speech enhancement in [233].

The use of large, open vocabulary continuous speech recognition (LVCSR) to substitute human listening evaluations is a recent innovation. For instance, an open source LVCSR system available from [153] was also used to evaluate TTS intelligibility in [234]. Previously, only closed vocabulary ASR had been used for transcription tasks,

as in [235]. Recently, ASR has also been used for other tasks in TTS such as the automatic selection of "clean" training utterances and speakers [236], and for transcription of training recordings in [166].

Little work has so far sought to establish the reliability of ASR for measuring TTS intelligibility. [21] found strong correlations between human word error rate (WER) collected from Amazon Mechanical Turk (MTurk) to the WER of 3 different ASR systems (IBM Watson, Google API and wit.ai). [237] also found correlations between MTurk, these ASR systems and MCD when building DNN-based TTS voices in Merlin. However, it remains unknown whether explicit ASR-derived rankings of multiple TTS systems correlate with those derived from paid, in-lab human transcribers.

4.3.2 Data

The Blizzard Challenge [104] provides evaluation data for the development of objective metrics [238]–[242]. The Blizzard Challenge is an annual event where participants are provided with a speech dataset for voice building and are asked to submit a defined set of synthetic samples for evaluation. The focus of the challenge changes from year to year; for example, samples were evaluated at varying noise levels in 2010, while the challenge was focused on Mandarin TTS in 2019. Each year, a large-scale human listening evaluation is conducted and participants submit samples of semantically unpredictable sentences (SUS) for a human transcription task that measures intelligibility. The test samples and evaluation data are available for download [1]. This resource was used to compare WERs computed using in-lab and online human transcriptions with objective ASR transcriptions. Specifically, I compare rankings of systems submitted to the Blizzard Challenge in 2011, 2012, 2013, 2016, 2017 and 2018. The relevant data for 2016, 2017 and 2018 was unavailable from the website and I worked with Blizzard Challenge organisers to format the results from the latter years for analysis. See the Blizzard Challenge summary papers [243]–[248] for more detail on and results from each challenge. The years 2014, 2015, 2019 and 2020 were excluded as these used languages other than English. I discuss the potential research directions with other languages in Sections 4.3.11 and 6.2.

Each year a section of the evaluation focuses on measuring the intelligibility of submitted systems. Paid listeners are recruited who type-in transcriptions in purpose-built sound booths under controlled conditions. These listeners are known as EP or

[1] The data is available from this link: `https://www.cstr.ed.ac.uk/projects/blizzard/data.html`

EE depending on the year of the challenge. Participating teams also recruit their own speech experts and online volunteers to conduct an evaluation. Known as *ES* and *ER* respectively, these are mainly composed of non-native speakers of English. In 2011, Amazon Mechanical Turk (AMT) was also used for evaluation.

Each year a new test set of SUS stimuli is submitted as well as the test sets of the previous two years. For each challenge 3 test sets were analysed in the ASR *maximum stimuli* sets (henceforth *Extra ASR*): 2011 (700 stimuli), 2012 (800), 2013 (900), 2016 (600), 2017 (600), 2018 (600). The data from the EH1 challenge for each year and for EH2 from 2013 were included. The test stimuli were created using a SUS generator and do not appear in the LibriSpeech dataset for the ASR system used (see below). The data were pre-processed to exclude punctuation, and all comparisons were made on upper-cased text.

Systems are randomly allocated a different anonymized letter each year. Some systems did not submit the 3 SUS test sets in a given year (such as system N in 2017) and those systems were excluded from analysis. System A is always natural speech but since recordings of the SUS sentences do not exist, they were not included in the analysis. Years 2017 and 2018 included systems based on neural text encoders and WaveNet-based vocoders, with earlier years including previous Unit Selection and SPSS-based TTS. Statistics were computed using the *Scikit-learn* Python package.

4.3.3 ASR Model

The ASR model was a pretrained LibriSpeech Transformer model available from ESpNet [153]. This had the advantages of being open-sourced, accessible and trained end-to-end (E2E) on a large (1,000 hours) multi-speaker corpus [249]. It performed with a WER of 4.9% on the LibriSpeech clean test set. As an E2E model, it had the disadvantage that extracting recognised phone strings to measure phone error rate (PER) was not possible, which may otherwise have offered insights into the reliability of ASR for TTS intelligibility. For example, [250] found ASR PER to be a superior means of TTS model selection than common loss functions. The reliability of using an E2E-model versus a hybrid model for TTS evaluation (in particular for implicit pronunciation modelling) could be an interesting topic for future work. For example, with what reliability could post-lexical pronunciations (e.g. *liaison*) be transcribed with phone recognition? Another interesting question also arises over the transcription of homophones (e.g. *shoes* and *shoos*): in [19], it was argued that a higher WER was

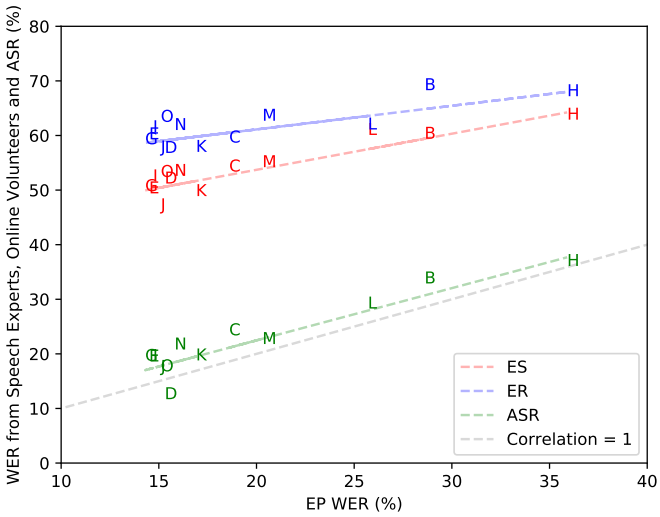

Figure 4.1: WER of systems for SUS stimuli used in the Blizzard Challenge 2018. The x-axis shows the WER for Paid Listeners (*EP*). The y-axis shows WER for the same stimuli from Speech Experts (*ES*), Online Volunteers (*ER*) and ASR (*ASR*). A linear regression line of best fit was added to aid visualisation for each evaluation type. The scales of x- and y-axis are different because the range of WERs was wider for *ES* and *ER* than *EP*. The dotted grey line is through the origin (0,0) and would represent correlation in WERs to *EP*. *ES* and *ER* demonstrated higher WERs than *ASR*.

observed for phone-based E2E models due to a less robust handling of homophones than graphemes.

4.3.4 Calculating WERs

For the Blizzard Challenge re-evaluation WER measured intelligibility as used in the Blizzard Challenge. As will be explained in Section 4.3.11, character error rate (CER) is more sensible since the E2E-ASR model operates at the character level. CER is used in further experiments later in this chapter.

For each system in the Blizzard Challenge data, the WER was computed across all stimuli used in a set, including all human and ASR transcriptions. For the human evaluation, any blank entries were disregarded from analysis. In-depth verification of human transcriptions was infeasible due to the number of transcriptions such a verification would involve. The WER was computed using the *fastwer* package.

4.3.5 Visualising WERs

In order to draw conclusions about the validity of using ASR for ranking TTS systems, an aggregate analysis of multiple systems across multiple years of the Blizzard Challenge would be required. An initial method of visualisation is shown in Figure 4.1 which shows results for the year 2018. Paid in-lab participants (*EP*) were treated as gold standard in terms of evaluation quality, since they provided the lowest WERs and performed the transcriptions in controlled conditions. The letters in the figure refer to the competing systems as they appeared in the Blizzard Challenge. The x-axis labels the WER obtained by systems when samples were transcribed by *EP*. The same system evaluated by all listener types always has the same x-coordinate. The y-axis presents system WERs according to the transcriptions produced by speech experts (*ES*), online volunteers (*ER*) and ASR (*ASR)*. The rankings by each listener type vary along the y-axis.

A perfect correlation of *EP* to a listener type would entail the same distance in magnitude along the x- and y-axes between systems. Ideally, the systems would be spread out along the grey line which would have exact correlation. The relative rankings by each listener type is indicated by the height difference between the letters. To aid visualisation, a linear regression fits for each listener type was also plotted.

Consistently all *ES* and *ER* scored higher WERs than *EP*. All *EP* were native speakers, but *ES* and *ER* conducted their test online and included non-natives. Noticeably, ASR consistently achieved lower WER than the *ES* and *ER*, a trend that is repeated in all years analysed (see Figure 4.2).

This method of WER visualisation did compare listener types but the letters were difficult to read (for instance the letters 'G', 'E' and 'I' for *ASR* in Figure 4.2) and observing the rank of systems vertically seemed counter-intuitive. Furthermore, a figure had to be created for each year and analysed separately. Instead, the WERs were averaged for each system in each year for the *EP*, *ES*, *ER*, *ASR* groups. This is presented in Figure 4.2 alongside *Extra ASR*.

Figure 4.2 shows the difference in WER from *EP* for each year analysed. Each listener type is denoted by colour and the scores are offset around a year label to aid visualisation. Each bar represents the mean and 2 standard deviations difference in WER from *EP*. The bars are colour coded according to the transcription method. The *ASR* bars in blue are the same stimuli as transcribed in the formal human evaluations, ranging between 25-40 stimuli depending on the year. As noted above, the *Extra ASR*

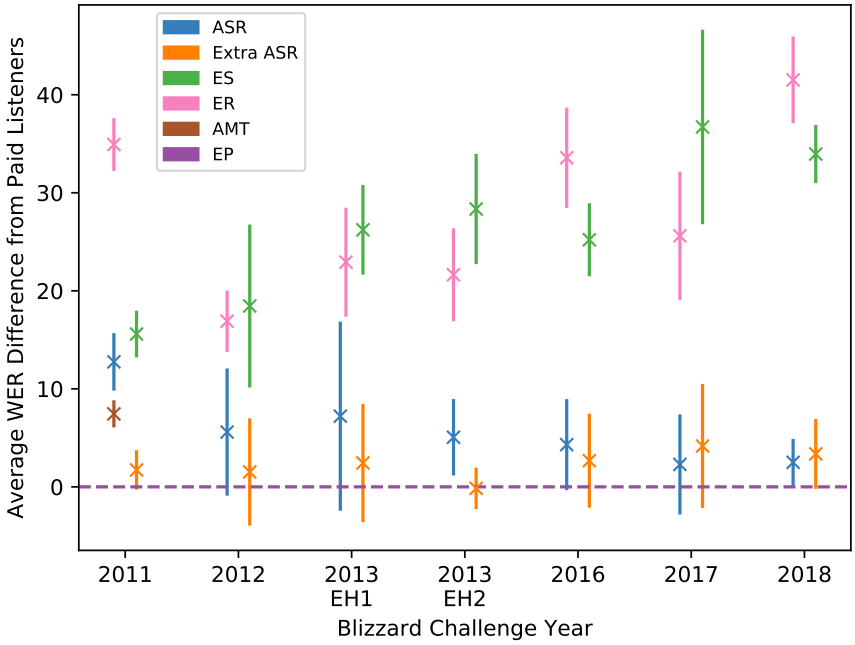

Figure 4.2: Aggregate Difference in WER from Paid Participants. Each bar shows the mean and 2 standard deviations for each listener type: Speech Experts (*ES*), Online Volunteers (*ER*), Amazon Mechanical Turkers (*MTurk*), the same stimuli ranked by ASR (*ASR*) and the maximum number of test SUS synthesised each year (*Extra ASR*)

bars in orange correspond to 3 SUS test sets (600-900 stimuli per year) Note, since *Extra ASR* contained additional stimuli, direct comparison between *Extra ASR* and the other methods was not possible. However, *Extra ASR* bars in Figure 4.2 were included to show that with more stimuli of the same genre, WERs remained similar between *Extra ASR* and *EP*.

ASR performance is close to *EP* in WER for every year except 2011, where *MTurk* achieved lower WERs. *ASR* gives consistently lower WERs than the *ES* and *ER*. The latter groups have high WERs as their evaluations are conducted more informally than for the *EP* and non-natives are consistently above 60% of listeners each year. Similar WER averages and spreads are achieved by the *ASR* and *Extra ASR* sets. This was as expected since the genre of text was similar. The mean of the *Extra ASR* bar for 2013 EH2 was below the mean of *EP*.

This Figure shows *ASR* achieved similar WERs to human participants when transcribing SUS stimuli from the Blizzard Challenge. Since this work, the authors of [22] too conducted a re-evaluation of multiple years of the Blizzard Challenge with a large scale MOS test. In their analysis, ASR was used as an objective metric alongside Signal-to-Noise ratio and MOSNet. ASR WER was found to exhibit the strongest

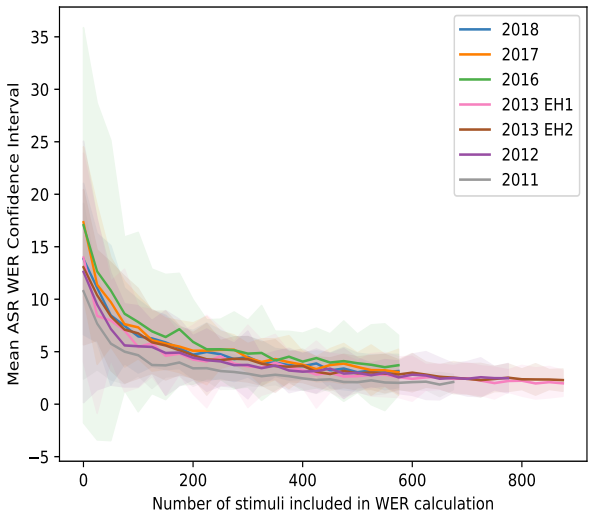

Figure 4.3: Bootstrapped WER confidence intervals averaged across *Extra ASR* stimuli. The bootstraps were conducted in steps of 20 stimuli. At each step, the mean and variance confidence interval in WER of all systems in a year were computed. A solid line represents the mean confidence interval for a year as stimuli were increased. The shaded bands represent 2 standard deviations in the confidence intervals of all systems in a year.

(negative) correlation to MOS scores out of the objective metrics tested. The low WERs from *ASR* and *Extra ASR* were encouraging, but questions still remained about the confidence and stability of these scores.

4.3.6 Bootstrapping ASR Confidence Intervals

WERs fluctuate across stimuli thus the confidence in the WERs come into question. Measuring confidence in the WER metric is thus important if it is to rank differences between TTS systems. In particular, can confidence intervals of WERs narrow as the number of stimuli is increased? During this analysis, a bootstrap method [251] for confidence intervals was attempted inspired by [252]. A bootstrap of ASR WER involves sampling WERs from a bag of stimuli to remove possible effects of sentence ordering on the metric. For simplicity, from Section 4.4 2 standard deviations were instead used as a confidence interval.

The bootstrap was successively run in steps of 20 stimuli up to the size of the *Extra ASR* test sets for each year. By using the Extra ASR set, confidence intervals for a

larger number of stimuli could be reviewed. For each step (of 20 stimuli) individual WER scores were resampled with replacement. 1000 model simulations of WER were computed in each step. The simulated WERs were sorted and the 95% confidence interval was plotted using the 25th and 975th percentiles at each step (2.5% either side of the distribution). These upper and lower bounds formed the bootstrapped confidence interval around the WER given a certain number of stimuli. The confidence intervals allowed visualisation of statistically valid differences between TTS systems and datasets as the number of stimuli under test increased.

Figure 4.3 shows the average WER confidence interval after bootstrapping. The confidence interval for each year is an average of the confidence interval of all systems at each step. The confidence intervals in a single year were averaged across all systems.

The lines are the mean interval at each step of 25 stimuli, the shaded area shows 2 standard deviations around the mean. The means begin to stabilise around 500 stimuli to around 4%. In 2016, the range of confidence intervals was more diverse than other years, but its mean score was similar to other years.

Narrowing confidence intervals show systems may be more reliably scored with *Extra ASR* stimuli. Below the effect of increased stimuli on significance testing is examined.

4.3.7 Visualising Significance in Rankings

4.3.7.1 Kendal-tau

Initially, rankings were compared from each listener type using the Kendall-Tau rank correlation statistic [253]. However, this statistic computes correlations on the raw rankings. Since many systems exhibited no significant differences between one another, the correlation coefficient and p-value were misleading. They would have been indicative if every system had a significant difference between itself and its neighbouring ranked systems.

4.3.7.2 Rankings by Pairwise Wilcoxon Signed-Rank Test p-Values

The Blizzard Challenge uses the pairwise Wilcoxon signed-rank test for significance. This statistic computes a matrix of p-values between systems in a ranked order. In a matrix, each cell is a p-value between a pair of systems. Figure 4.4 simplifies two such matrices using a p-value threshold of 0.005 where blue indicates a significant pair, while red represents pairs above the threshold.

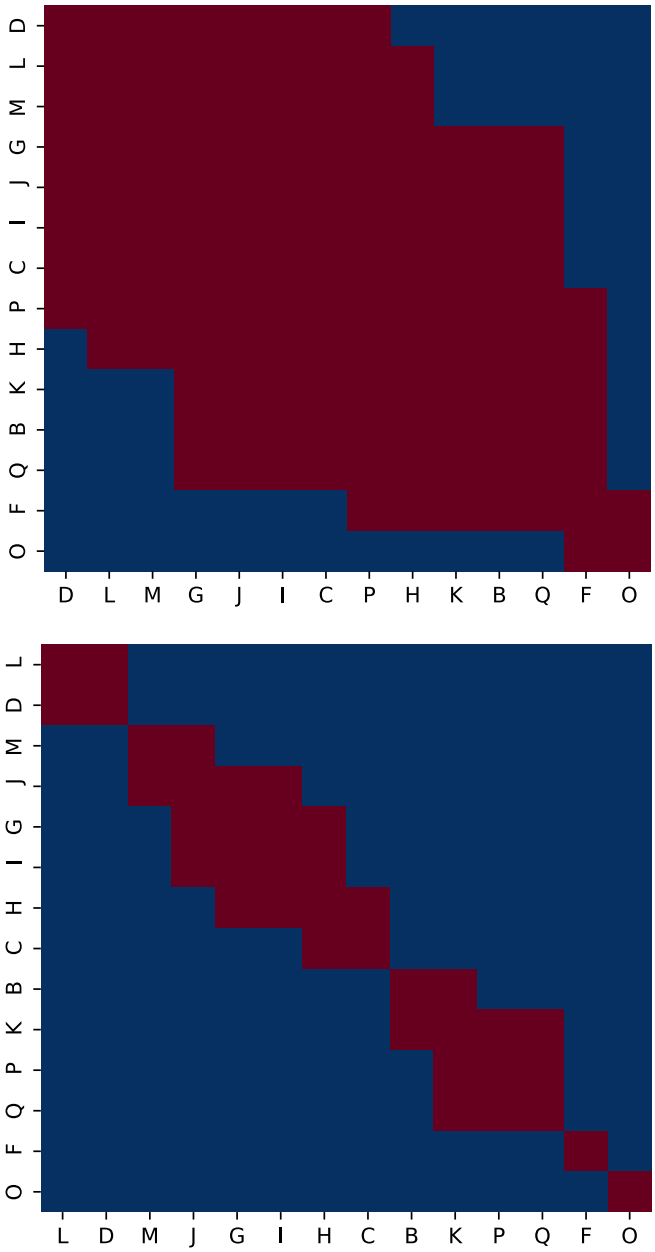

Figure 4.4: Heatmap of pairwise p-values for systems ranked by *EP* with 25 stimuli (Top) and *Extra ASR* (600 total - bottom) for the Blizzard Challenge 2017. Blue indicates a p-value below 0.005 between the pair. Red indicates no-significance has been identified. Systems appear along axes in order according to transcription method (*EP* on top, *Extra ASR* on bottom). With extra stimuli p-values were lower.

Year	Paid Listeners	ASR	Extra ASR
2018	G E I O J D N K C M L B H	D J O E I G K N M C L B H	G D N O E C J I K M B L H
2017	D L M G J I C P H K B Q F O	L D M P I J G H C B Q K F O	L D M J G I H C B K P Q F O
2016	F L C M G B I O J K E P H N	F L C K I G B M O J E H N P	F C L G K M I B O E J P H N
2013 EH2	C I K F M H G L O E N D J	I C K L H M O G E F N D J	I C H M K G L F E N O D J
2013 EH1	C I M H L K F N P	I C K H M F L N P	C M H K L F N P
2012	C I H F D B G K J E	H C D B K I F G E J	H C D F B I G K E J
2011	F G M D C K E L H J B I	M C D F G E H I L B K J	C D F M G H E K I B L J

Figure 4.5: Groups of no-significance for Paid Listeners (*EP*), ASR and *Extra ASR*. Each line represents a unique grouping of systems as found in the p-value heatmaps. For the Pairwise Wilcoxon signed-rank test, the significance level was set at a p-value of 0.005. When stimuli only used in the formal evaluation were included (Paid Listeners and ASR), the groups of no-significance encompass more systems than with extra stimuli (*Extra ASR*). *ER*, *ES* and *MTurk* groupings were omitted from this Figure due to space considerations and ease of visualisation - these also had long groups of no-significance such as the Paid Listeners and ASR. Note that systems which did not have 3 SUS test sets available were excluded from analysis.

Aggregate statistics using the pairwise Wilcoxon signed-rank matrices for each listener type were computed. The top heatmap shows the rankings and p-values for *EP* in 2017. The bottom one shows the ranking and p-values according to *Extra ASR*. The order of systems along the axes represent the ranks by the transcription method in question (e.g. *EP* on the top and *Extra ASR* on the bottom).

The heatmaps display overlap of no-significance between systems with partial rows of red cells. The less red in the heatmaps, the finer the significance between the systems. I sought to visualise the groups of no-significance across the systems. Groups of no-significance were extracted programmatically. Overlapping bubbles were attempted as used to report results for the voice conversion challenge (e.g. Figure 3 in [254]). However, the number of bubbles to demonstrate each overlapping group of no-significance made the figures illegible. Furthermore, a figure for each listener type would have been necessary and it would have been difficult to visualise effects across years.

Instead, the rankings were tabulated and lines were plotted for each overlapping group of no-significance as shown in Figure 4.5. Each of the unique partial rows of no-significance in the heatmaps translate to a blue line in Figure 4.5. The first partial row in the heatmaps above spans systems D to P (the top line in Figure 4.4), the second partial row spans systems D to H. Hence these are the first two lines in the 2017 Paid Listeners (*EP*) cell in Figure 4.5. For each challenge 3 rankings are shown (*EP*, *ASR* and *Extra ASR*).

Consistently, the rows of no-significance are similar for *EP* and *ASR* but longer than in *Extra ASR*. Figure 4.5 consistently shows that *Extra ASR* indeed found similar significance groups to *EP* and more significant differences when stimuli were increased.

Figure 4.5 also shows that some systems which are further than 1 step away in a rank may be in a similar group of no-significance. For example, system Q in the top line of *ASR* 2017. Although the mean performance of a system gives a particular ranking, the spread in its performance might result in no statistical significance when tested. The mean score of system Q was skewed by 2 low quality outlier stimuli. Such stimuli may be very important to examine for systems in deployment, and wide variance is also observed when using increased stimuli with higher confidence such as with system I in *Extra ASR* 2018 and with systems C and E in *Extra ASR* 2011. The problems resulting from smoothing out the effects of certain individual stimuli was the focus of [177] where the authors proposed evaluating systems on samples with the largest differences in audio output. Alternately, differences in systems may be revealed

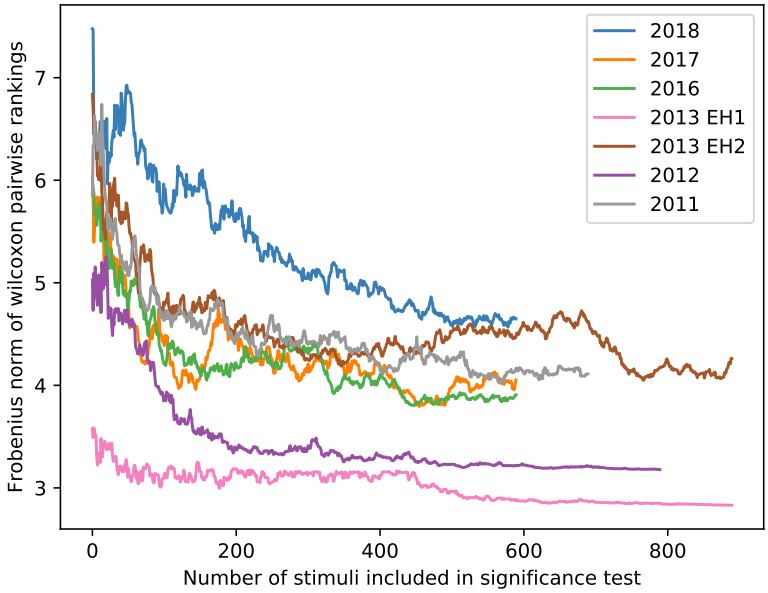

Figure 4.6: Frobenius norm of pairwise Wilcoxon rankings varying according to number of stimuli included in significance tests. The desideratum was a convergence level for each curve. This level would demonstrate the number of stimuli where the amount of discovered significance was optimised with as few stimuli as possible.

using targeted stimuli as demonstrated in [111], and as argued throughout this thesis.

4.3.7.3 Frobenius norm of p-value matrices

How did significance levels change as the number of stimuli under consideration was increased? Could an optimum point be found where significance was maximised with as few stimuli as possible? To visualise how significance varied, the Frobenius norm was calculated of each pairwise Wilcoxon p-value matrix as the number of ASR stimuli was increased. The Frobenius norm is the square root of the sum of all the squared values of a matrix. In Figure 4.6 the Frobenius norms are plotted as the stimuli for each challenge were increased.

The falling curves reflect falling p-values overall. There is a fall for all challenges, but to a differing degree for each. The absolute value of the norm is dependent upon the total number of systems (e.g. 2013 EH1 contained the fewest systems and has the lowest Frobenius norm curve). More noteworthy is the relative gradient change for each curve and to find where they converge - this level indicates where optimal significant differences between systems in a challenge were found.

The curves fall the most in the first 200 stimuli. 2013 EH1 falls further after 400 stimuli when reaching a subset of the *Extra ASR* stimuli. Each curve has its own relative convergence level arising from the performance on the *Extra ASR* stimuli, the number of systems, and the relative quality of each in a challenge. Levelling can be observed from between 400-800 stimuli, although this is less clear for 2013 EH2 where the curve increases after 400 stimuli until it drops further around 700 stimuli. The test stimuli included can effect whether a convergence level is found.

4.3.8 Summary of Blizzard Challenge Re-evaluation

1. *How does ASR compare to paid listeners when transcribing synthetic speech in the Blizzard Challenge?*

As shown in Figure 4.2 *ASR* and *Extra ASR* obtain WERs that are between 5-10% higher than paid listeners (*EP*) for SUS stimuli in English. Performance on other languages would be interesting future work.

2. *How do confidence intervals over ASR WERs change as the number of TTS test stimuli is increased?*

As shown in Figures 4.3 and 4.4 confidence intervals narrow and p-value estimates decrease as the number of stimuli is increased. This is an advantage over human listeners as more stimuli can be automatically evaluated with a resulting higher statistical confidence in the results obtained.

3. *Do ASR transcriptions identify the same significant differences between system pairs as the paid listeners?*

As shown in Figure 4.5 *EP*, *ASR* and *Extra ASR* identify similar groupings of no significance across multiple years of the SUS stimuli.

4. *Are there any benefits to increasing the number of stimuli for ASR transcription?*

   The frobenius norm was used to find the plateaux in p-values as the number of stimuli were increased. Figure 4.6 shows finer-grained significant differences can be obtained with increasing stimuli, sometimes to a plateau. Were the *EP* transcriptions available it would be interesting to conduct similar analyses for other text genres (e.g. news).

4.3.9 Conclusions

The above analysis showed that ASR performed reliably for evaluating intelligibility of TTS systems in the Blizzard Challenge - indeed on a comparable level to the challenge's paid listeners (*EP*). Using increased stimuli, *Extra ASR* also detected more statistically significant differences between pairs of systems, which would have been expensive to find in the human evaluations. ASR can be a reliable and convenient metric to measure intelligibility of SUS sentences in the Blizzard Challenge, as long as a sufficiently large number of stimuli are used.

Demonstration of similar significant groupings in the Blizzard challenge may validate the use of E2E-ASR to a degree, but there remain further questions regarding its reliability, in particular the reliability of transcription of words exhibiting difficult G2P relations.

4.3.10 Further Research Questions

1. What are the qualitative differences between ASR and human-transcription?
2. What kind of transcription errors does E2E-ASR make?
3. How fair is the evaluation of pronunciations of difficult G2P words by E2E-ASR?
4. Are significant differences observed between input-types when targeted stimuli are used at scale?

There may remain further questions from the reader such as what is the effect of implicit language modelling or text genre/expressiveness in transcription? Due to the exponential avenues to ponder, the rest of this chapter intends only to provide insight toward the questions above.

4.3.11 Human/E2E-ASR Transcriptions and the CER

EP was considered gold-standard in the previous analysis but its imperfections should still be noted as the use of humans is inherently subjective: humans get tired, can be inconsistent, can disagree, or not transcribe properly. Furthermore, in online tests such as crowdsourcing, the researcher has little control over the audio environment of a listener - such as differing levels of background noise or their equipment (earphones, headphones, laptop speakers).

Nevertheless, ASR also exhibits imperfections as a method of transcription - particularly for differences in text normalisation when evaluating purely on text. For instance, word compounds can be represented as one or two words: e.g. *beerman* or *beer man*. Also, ASR may recognise the wrong word out of a pair of homophones (*boil*/*boyle*, *eyeblinks*/*iblinks*). There may be differences in spelling too: in one case it was observed the ASR model outputted *foretel* with only one *l*, while the reference text contained two: *foretell*. ASR in deployment may undergo Inverse Text Normalisation (ITN) - reformatting of characters to readable text (see [255], [256]) which may help with some of these issues but which was not implemented here. Text normalisation differences between transcriptions and reference text could artificially inflate WER. Furthermore, E2E-ASR possesses an imperfect pronunciation model that is prone to mis-recognise non-standard words and difficult G2P (or P2G) words such as foreign words and proper names. Attempts to improve pronunciation modelling of difficult words in E2E-ASR include [257]–[259]. For this reason, it was more sensible to use CER to avoid penalising minor spelling differences at the word level. CER offers a proxy for PER to a closer degree than WER. The rest of the ASR results presented in this chapter therefore use the CER.

Another noteworthy difference between human and ASR transcription for TTS evaluation is the effect of babbling on WER scores as insertions. The attention mechanism in S2S-TTS models can fail with the consequence of babbling audio. This is a problem that has been tackled in works such as [16], [260]. When an attention failure occurs, the ASR outputs a random string of words and characters. For example:

Reference Text:

FOR AGE RELATED HEALTH PROBLEMS

ASR on TTS Babble:

FOR AGES RELATED HOW A H M H M H M H M H M

The large number of insertions skewed the error rate across a test set considerably. By checking for outlier-lengths of text-strings in all transcriptions below, all cases of babbling were excluded from the CER calculations. The considerations above deserved mention now before proceeding to evaluate the Tacotron systems below.

4.4 Analysis for Pronunciation Evaluation

The objective of transcribing the following test sets was to assess the reliability of E2E-ASR transcription for words with difficult G2P relations. Simultaneously, this section also compares the CER of text- and phone-input to Tacotron when synthesising such words.

4.4.1 Systems

The rest of this chapter explores ASR transcription to evaluate input-types (`G` and `P`) to Tacotron. The same E2E-ASR model was used as in the Blizzard Challenge re-evaluation. This model transcribes in English and was trained on a large corpus with multiple speakers. It would be interesting to evaluate TTS in French on a larger scale but an E2E-ASR model trained on French data was unavailable. The English systems from Chapter 3 were used: `G`, `P`, `GM` and `PM`.

4.4.2 Test Set Description

The two sets described here were created to test the pronunciation of words with difficult G2P relations. These contained 3,000 words each. Originally 6,000 words were used, however this was decreased due to time considerations as these sets were used with 28 systems in Chapter 5.

1. *In-LJ*. 3,000 words were randomly selected from LJ speech inserted into the carrier sentence "*Now we will say ... again*".

2. *Out-LJ*. 3,000 words were selected that were mispronounced by the `LJ Token` G2P model from Chapter 2. Each word was inserted into the carrier sentence "*Now we will say ... again*." This set contained words with difficult G2P relations and words of multiple morphemes.

The *In-LJ* and *Out-LJ* sets were recorded with natural speech (`N`). Due to social distancing policies, they were recorded in a quiet, consistent environment (home-office) with a high quality microphone. Each content word was recorded and then artificially sandwiched within a single recording of the carrier sentence "*Now we will say ... again*".

Reference Transcriptions

Index	1	2	3	4	5	6
Reference 1	NOW	WE	WILL	SAY	RESULT	AGAIN
Reference 2	NOW	WE	WILL	SAY	RURAL	AGAIN
Reference 3	NOW	WE	WILL	SAY	BEERMAN	AGAIN

Raw E2E-ASR Hypotheses

Substitutions/deletions in carrier sentences

Index mismatches

Index	1	2	3	4	5	6	7
Hypothesis 1	NOW	WE	WILL	SAVE	RESULT	AGAIN	n.a
Hypothesis 2	NOW	WELL	SAY	RURAL	AGAIN	n.a	n.a
Hypothesis 3	NOW	WE	WILL	SAY	BEER	MAN	AGAIN

Normalised E2E-ASR Hypotheses

Ignore substitutions

Capture 5 to -2 and remove whitespace

Index	1	2	3	4	**5-> -2**	-1
Hypothesis 1	NOW	WE	WILL	SAVE	RESULT	AGAIN
Hypothesis 2	<TOK>	NOW	WELL	SAY	RURAL	AGAIN
Hypothesis 3	NOW	WE	WILL	SAY	BEERMAN	AGAIN

Insert artificial tokens to normalize length of carrier preamble

Compute CER with index 5 from Reference Transcriptions

Figure 4.7: Normalisation of E2E-ASR hypotheses for keyword spotting of content token(s). Tokens were delineated by whitespace but this was problematic for substitutions/deletions in the carrier sentence. For consistency, the length of the carrier preamble was normalised with artificial tokens and any tokens between indices 5 and -2 were used. Whitespace was removed from the captured indices. This method did not handle where *say* or *again* were merged with the the content token(s).

4.4.3 CER with Targeted Stimuli

The *In-LJ* and *Out-LJ* sets required additional preprocessing for evaluation. To control for differences in the recognition of the word tokens in the carrier sentence, the CER was computed on the content token(s) in each sentence rather than on the entire sentences as with the SUS test sets in the Blizzard Challenge. However capturing only the content token(s) was not straightforward.

ASR would sometimes mistranscribe aspects of the carrier sentence either as a substitution (e.g. *SAVE* or *SEE* for "SAY"), or a deletion (e.g. *WELL* for "WE WILL", *WOOZY* for "WE WILL SAY"). Occasionally the predicted word merged with the preceding word "SAY" or the final word "AGAIN". For instance "NOW WE WILL SAY LERWICK AGAIN" was transcribed *NO WOOL SAILOR RICK AGAIN* and "NOW WE WILL SAY PENTHOUSES AGAIN" was transcribed *NOW WE WILL SAY TEN THOUSAND*. It was not fair to compute CER on the entire sequence due to examples such as these, thus it became necessary to compute CER on the content token(s) only.

The issue is akin to keyword spotting [261]–[264]: how to extract a word from a longer sequence of audio. A keyword spotting method could have been used here to extract the content token(s) but such a method would not have solved the problem of merges with surrounding words such as *SAILOR RICK* above. Indexing based on whitespace was a simpler, heuristic approach. This approach did still require the transcription of the carrier sentence to be as consistent possible, but it had the advantage over other key-spotting methods of simplicity.

Deletions were most problematic because the indices of the content tokens would change and a deletion of "AGAIN" would put the content token at the end of the sentence. Therefore, a solution was to artificially add tokens at the start of the sentence until each contained at least 6, and then to evaluate the content-portions after the first 4 tokens and the penultimate token. This process is illustrated in Figure 4.7. The penultimate index could not simply be used since there may have been spaces in the transcription of the content (e.g. "BEERMAN" and *BEER MAN*) which would have gone uncaptured.

All content tokens in the reference sentences were single words without whitespace. To ensure content tokens with whitespace were included, all whitespace was removed from the content-portions after the first 4 words and before the penultimate word.

While this approach may still be biased against examples where "SAY" or "AGAIN"

were merged, a more complicated approach would not have solved this problem of merges. As will be seen by the consistency with which the target tokens were captured in examples below, the heuristic approach taken here overall worked well. By adding artificial tokens to the front and only taking tokens between the fifth and penultimate token, the content word was still captured consistently. The approach is illustrated in Figure 4.7.

4.4.4 Results

4.4.4.1 Natural Speech (N)

Let us analyse the differences between *In-LJ* and *Out-LJ* with natural speech (N) first to gain an understanding of the reliability of E2E-ASR transcription on gold-standard speech. The CER for *In_LJ* was 7.3 $\pm$ 3 and for *Out_LJ* was 14.8 $\pm$ 5.4. The CER was higher for *Out-LJ* than for *In-LJ* notably but not significantly according to the two standard deviations.

Table 4.1 shows example errors in both sets. The highest CERs for both sets came with words mistranscribed (near-) homophonically. For instance from the *In-LJ* set the word "CIRCULATORY" was transcribed as *A CIRCULAR TREE* and "DISMAYED" was transcribed *THIS MAID*. More examples of near-homophonic transcription are shown in the *In-LJ* columns. For *Out-LJ*, *DO SIT AS AN EYES* was observed for "DECITIZENIZE" and *WHO LOOK IN EYES* was observed for "HOOLIGANISE". Notably for *Out-LJ*, certain foreign words and proper names were also mistranscribed (like "BODEGA", "BOTSWANA", "RHODESIA"). Understandably E2E-ASR transcription is less reliable for the *Out-LJ* than the *In-LJ* set. The scores were consistently higher for the *Out-LJ* set for all TTS systems as well.

The test words placed in carrier sentences are out of context, which in part may be a cause of ASR mistranscription and arguably may not reflect a real-world use case. However, what of E2E-ASR as a relative measure of pronunciation? Although this method of transcription is unreliable in absolute terms, on aggregate what are the differences between the systems? With a high number of stimuli, can some proxy of pronunciation still be inferred amongst mistranscriptions?

4.4.4.2 Reliability for E2E-TTS pronunciation evaluation

Since E2E-ASR transcription is unreliable for natural speech, how reliable could it be to evaluate pronunciations by E2E-TTS systems? In the rest of this chapter, I analyse

In-LJ		Out-LJ	
Reference	**N Hypothesis**	**Reference**	**N Hypothesis**
HOURLY	NOW I BE	DECITIZENIZE	DO SIT AS AN EYES
ANATOMY	AND AFTER ME	BOTSWANA	WHAT ONE EARTH
DISSOLVED	IT IS OLD	HOOLIGANISE	WHO LOOK IN EYES
CIRCULATORY	A CIRCULAR TREE	HOMEOMORPHIC	WHOM I AM MORE FIT
DISMAYED	THIS MAID	CENTENARY	SEVENTEEN EIGHTY
ADEPTS	THE DEPTHS	BODEGA	THE DAGGER
BUOYED	BOYS	BINOMIALLY	BY NO MEANS
LIED	LIGHT	BICENTENNIALS	BY SENTINELS
MORPHOLOGICAL	MORE PHILOLOGICAL	SCHIZOPHRENICALLY	SKITS ARE PHRENICALLY
MACCLESFIELD	MACKELSFIELD	BAGATELLE	BACK TO TELL
KAY	K	REGIONALIZE	REGION LESS
TENTS	TENSE	RHODESIA	RADIO
RESIDENCE	RESIDENTS	UNCATALOGUED	UNCANNY GLOGGED
MISFIRES	MISS FIRES	EMOTIONALIST	THE MOTION LIST
CRISSCROSSING	CHRIS CROSSING	OBSTRUCTIONISTIC	OBSTRUCTION IS STICK
DYE	DIE	UNCOMFORTABLEST	UNCOMFORTABLE LIST
INN	IN	WELFARISTS	WELL FAIREST
AFFAIR	A FAIR	PRONOUNCEABLE	PRONOUNCIBLE
TAILORS	TAYLORS	CONSCIOUSNESSES	CONSCIOUSNESS IS
FAMILIARIZE	FAMILIARISE	INSTITUIONALIZING	INSTIUTIONALISING

Table 4.1: Example transcriptions from the *In-LJ* and *Out-LJ* sets with natural speech. Note: whitespace was removed from hypotheses before calculating CER. Errors in both sets arise with words that are transcribed with alternative characters homophonically (*CIRCULATORY* for "A CIRCULAR TREE", *THIS MAID* for "DISMAYED"). *Out-LJ* words exhibit mistranscription with unusual character contexts in English such as "BOTSWANA", "BOGEGA" and "RHODESIA". E2E-ASR transcription of words with difficult G2P contexts is unreliable.

Reference	G	P	GM	PM
DEMEAN**OR**	DEMEANOUR	DEMEANOUR	DEMEANOUR	DEMEANOUR
P**EA**L	PEEL	PEEL	PEEL	PEAL
REGULARI**S**ATION	REGULARIZATION	REGULARIZATION	REGULARIZATION	REGULARIZATION
AUSTRALIA**N**	AUSTRALIA	AUSTRALIA	AUSTRALIA	AUSTRALIA
INSTITUTIONALI**S**ING	INSTITUTIONALIZING	INSTITUTIONALIZING	INSTITUTIONALIZING	INSTITUTIONALIZING
LEFT**IST**	LEFTUS	LEFTUS	LEFTUS	LEFTUS
BET**AS**	BETUS	BETUS	BETUS	BETUS
HOPWOODS	UPWARDS	UPWARDS	UPWARDS	UPWARDS

Table 4.2: Some examples of consistent mistranscriptions across 4 TTS systems. The differences in spelling are in bold under the Reference column. Each TTS system had the same difference in spelling from the reference. Although mistranscriptions are common in the *Out-LJ* set (which penalise `P` and `PM`), consistencies in mistranscriptions permit a relative comparison between TTS systems when using a large number of stimuli.

the transcription of these sets by the English TTS systems from Chapter 3.

The errors from *N* showed that E2E-ASR mistranscribed (near-) homophonically for words which do not have a clearly defined homophone partner. For instance, DECITIZENIZE and *DO SIT AS AN EYES* may not traditionally be thought of as homophones but their inferable pronunciations are very similar. Furthermore, (near-) homophonic pairs were observed to be consistently mistranscribed across TTS systems. Examples of these are shown in Table 4.2. For instance, "PEEL" was mistranscribed as *PEAL* consistently by `G`, `P`, `GM` and `PM`. The relative transcriptions between systems may therefore offer further insight into the reliability of E2E-ASR transcription. If `G` is to be prone to mispronounce words from the *Out-LJ* set, should not the transcriptions of `G` obtain higher CERs than `P`?

I observed a problem between the relative transcriptions of systems *G* and *P* for some particular words where learning the pronunciation from character contexts would be insufficient to learn the correct pronunciation. Some examples are shown in Table 4.3 from similar systems that appear in Chapter 5. Mispronunciations of difficult G2P words can be masked by E2E-ASR mistranscription. If a word spelling exhibits difficult or rare G2P relations, then a mispronunciation may actually be transcribed at the character level correctly. Or likewise, a correct pronunciation for a word with unusual G2P relations may not be correctly transcribed, thereby putting `P` at a disadvantage.

Let us look at examples. Table 4.3 shows mispronunciations by a TTS system trained on text-input (`grapheme-only` from Chapter 5) that received equal or lower CERs than correct pronunciations by a system trained with phone-input (`Trigram-500`

- trained with mixed representations but using phone-input at test time). The reader is encouraged to listen to the corresponding audio samples online[2] for the examples presented in the rest of this chapter.

For instance, "CRINGING" was pronounced with the voiced velar nasal [ŋ] by `grapheme-only` instead of the dental nasal and affricate [ndʒ] by `Trigram-500`. The mispronounciation was transcribed with the characters *NG* which masked the pronunciation error. Although "*CRINGING*" was mispronounced by `grapheme-only`, both `grapheme-only` and `Trigram-500` received 0 CER. This illustrates how an E2E-TTS system with text-input may have mispronunciations unnoticed with E2E-ASR transcription. "OBESENESS" is another example where both `graphemes-only` and `Trigram-500` have the same transcription but the pronunciation of `grapheme-only` was correct despite a mispronunciation.

In "COXSWAIN" and "SALISBURY", the correct pronunciation by *Trigram-500* obtains a higher CER than the same words mispronounced by `grapheme-only`. These samples illustrate how E2E-ASR simultaneously lets mispronunciations from a TTS system with text-input go undetected while penalising a better pronunciation of a word with difficult G2P relations that is not "readable".

The pronunciation of "COXSWAIN" by `G` was [kɑːkswɛɪn] instead of the correct pronunciation by `P`: [kɑːksən]. However, the transcription for `G` was *COX WENT* whereas for `P` it was *COXON*, with a higher CER. "SALISBURY" is another example.

In essence, this E2E-ASR system presents a biased metric in the evaluation of pronunciations by text- and phone-based TTS systems. In practice, pronunciation evaluation via characters is unfair because mistranscriptions by E2E-ASR potentially mask mispronunciations and penalize correct pronunciations.

Another issue with the relative comparison between systems is that sometimes both transcriptions may be wrong, thus it can be difficult to compare the pronunciations by the TTS systems. For instance, in Table 4.4, the word CHAMOMILE is transcribed as *CHUMMY* for `grapheme-only` and *CANNIBAL* for `Trigram-500`. Both CERs were high, neither resemble the reference transcription, even though the pronunciation of `Trigram-500` is more adequate. Other examples of high CERs are given in OSCILLOGRAPHS and SUGARCANE. While differences in CER on aggregate may be observed between TTS systems, ultimately these numbers are not reliable in absolute terms for the evaluation of difficult pronunciations.

[2]Samples are available here: `https://homepages.inf.ed.ac.uk/s1649890/chap4/`

Reference	grapheme-only Pronunciation	grapheme-only Hypothesis	grapheme-only CER	Trigram-500 Pronunciation	Trigram-500 Hypothesis	Trigram-500 CER
CRINGING	[krɪ**ŋŋ**ɪŋ]	CRINGING	0	[krɪ**n**dʒɪŋ]	CRINGING	0
OBESENESS	[əbɪ**z**nəs]	A BUSINESS	33.3	[oʊbisnəs]	A BUSINESS	33.3
COXSWAIN	[kɑ:ks**wem**]	COX WENT	57.1	[kɑ:ksən]	COXON	80
SALISBURY	[sɔlɪ**z**bəri]	SALISBURY	0	[sɔlzbəri]	SOLSPIRY	50

Table 4.3: Masked pronunciation errors from `grapheme-only` and mistranscription of better pronunciations by `Trigram-500` (a mixed representation system using phones). Pronunciations are given in broad IPA and errors are shown in bold. White space was removed before caluclating CER. Errors from `graphemes-only` can be masked by evaluating CER on text, as shown by the lower CERs for `graphemes-only` than `Trigram-500`. E2E-ASR transcription for pronunciation evaluation penalises some correct word pronunciations. The reader is once again encouraged to listen to audio samples.

The conclusion that CER is unreliable for this purpose is potentially important to a work such as [4] where the authors use ASR CER to compare grapheme- and phone-input to their system.

4.4.5 TTS System Results

What about the relative differences in CER scored between TTS systems? The results for `G` and `P` are shown alongside `N` in Table 4.5. `G` scored higher CERs than `N` and `P` for both sets but the relative increase in CER from *In-LJ* to *Out-LJ* was higher for `G` than `P` and `N`. The CER increased by 20.2% for `G` and but by only 13.6% for `P`. To understand the differences between the outputs better, Table 4.6 shows pronunciation errors made by the `G` system[3]. While the pronunciation of `P` was more intelligible than `G`, there were still mistranscriptions for `P` as in "PHARMACOPEIA". The transcriptions for `G` were less similar to the reference. Despite the imperfections of ASR transcription (in particular the favourable bias from masking mispronunciations) `G` scored a relatively higher CER than `P` for *Out-LJ*. Partly this is due to incorrect implicit G2P generalisation (e.g. "FOGGINESS" is pronounced with the [dʒ] sound instead of [g] sound, or the start of "SCHIZOPHRENICALLY" is pronounced with a [ʃ] sound instead of correct [*k*] sound. However, with the `G` sounds would often also be skipped, leading to high CERs. For example, "PHARMACOPEIA" was pronounced [fɑrməkəpəʃ]

[3]More TTS samples are available at this link: `https://homepages.inf.ed.ac.uk/s1649890/chap4/more_samples.html`

Reference	`grapheme-only` Pronunciation	`grapheme-only` Hypothesis	`grapheme-only` CER	`Trigram-500` Pronunciation	`Trigram-500` Hypothesis	`Trigram-500` CER
CHANEL	[**tʃeɪ**nəl]	CHANNEL	14.2	[ʃænəl]	SHINAL	50
CHAMOMILE	[**tʃ**æməmaɪl]	CHUMMY	83.3	[kæməmaɪl]	CANNIBAL	87.5
FALSIFIERS	[fɑlsɪ**fɪə˞**z]	FALSIFERS	11.1	[fɑlsɪfaɪə˞z]	FALL THE FIRES	50
OSCILLOGRAPHS	[ɑ:**skɪ**ləgræfs]	OSCAR THE GREX	75	[əsələgraefs]	SOLID RAFTS	80
SUGARCANE	[**sɪ**gɑ:keɪn]	THE BARKING	70	[ʃʊgəkeɪn]	TO HER KING	77.8

Table 4.4: Masked pronunciation errors from `grapheme-only` and mistranscription of better pronunciations by `Trigram-500` (a mixed representation system using phones). Pronunciations are given in broad IPA and errors are shown in bold. White space was removed before caluclating CER. Errors from `graphemes-only` can be masked by evaluating CER on text, as shown by the lower CERs for `graphemes-only` than `Trigram-500`. E2E-ASR transcription for pronunciation evaluation penalises some correct word pronunciations. The reader is encouraged to listen to audio samples at the aforementioned link.

which is nonsensical. Consequently there is a high CER (75%) for the transcription: *FARMER PUSH IT*.

The example presents a further issue of using E2E-ASR as a metric for pronunciation modelling. One cannot tell whether a high CER is due to character skipping/unstable speech or a G2P generalisation error. In the targeted stimuli evaluation for Deep Voice 3 [111], such errors were manually labelled and errors were categorised according to *mispronunciations*, *skipping* and *repetitions*. The nuanced categories of error are lost with E2E-ASR CER alone.

Although we see differences according to statistical significance across the sets, it is unclear whether at scale these differences are due to mispronunciations by the TTS systems or mistranscriptions by the E2E-ASR system. The metric is not reliable enough to be certain at scale, despite numerous examples presented in in Table 4.6.

Table 4.7 shows the CER for *In-LJ* and *Out-LJ* with systems including morphology. These systems performed with lower CERs for both sets than `G` and `P`. `GM` performed significantly better than `G` and `P` on *In-LJ*, showing that with morphology this TTS system has lower CERs. The columns of Table 4.8 show differences in transcription with the use of morphology.

With text-input, intelligibility of words with multiple morphemes was improved in both sets. `G` did not learn pronunciation contexts in longer words as well as `GM` as shown. For instance, "MAGNIFYING" was pronounced without the plosive [g] and the transcription was: *MAY NOT FIND*. These are nuances in pronunciation which

Dataset	N	G	P
In_LJ	7.3 ±3	30.7 ±3.2	17.9 ±3.4
Out_LJ	14.8 ±5.4	50.9 ±4.2	31.5 ±5.4

Table 4.5: CER results for *In-LJ* and *Out-LJ* for N, G and P. The CERs for *Out-LJ* are consistently higher than for *In-LJ* across all TTS systems. However, the gaps are larger for TTS systems than N and the gap is larger for G than for P. When testing with difficult words, G was less intelligible than P by a larger margin.

Reference	G Hypothesis	G CER (%)	P Hypothesis	P CER (%)
ANTIPODES	AND IT PUT IT	60	ANTIPATHIES	36.4
APPLETON	APLATOON	37.5	APPLETON	0
AZALEA	AND SLEEP	62.5	AZALEA	0
BAGATELLE	THE DEVIL	100	ABOUT TO TELL	63.6
BICENTENNIALS	THE TENANTS	80	BY SAINT DENIALS	42.9
BINOMIALLY	BENIGNLY	62.5	BY NOMILIA	44.4
BODEGA	A BODY	80	GOODBYE	71.4
EXTRADITE	EXTRICATED	40	EXTRADITE	0
FOGGINESS	FUDGES	83.3	FIVE MINUTES	63.6
HUMIDIFIED	WHO MAY FIGHT	72.7	HUMAN FIGHT	60
MEPHISTOPHELES	MISTER PHILIS	58.3	MEPHISTOPHOLES	7.1
PHARMACOPEIA	FARMER PUSH IT	75	FARMER COPEER	50
REGIONALIZE	ORIGINALISE	36.4	REGIONALISED	16.7
RESALABLE	REASONABLE	30	RESALABLE	0
SACERDOTAL	SIR TO PAUL	66.7	SASSER DOUGLASS	50
SCHIZOPHRENICALLY	SHES FREINDLY	83.3	SKITS OF FRENICALLY	35.3
SUNDIALS	SUNDERS	42.9	SUN DIALS	0
UPHOLSTERS	OF HOLSTERS	20	UPHOLSTERS	0

Table 4.6: Sample of transcriptions with higher CER for G than P. Note CERs are presented in percentages and it is possible to surpass 100% where the length of the reference is longer than the hypothesis. CERs were computed without whitespace. The pronunciations are intelligible for P but mistranscription causes a positive CER. Importantly, the CER was still higher for G than P consistently as in "MEPHISTOPHOLES" and "SCHIZOPHRENICALLY". Word pronunciations are still approximately inferred using CER despite imperfect transcription. The mispronunciation of "UPHOLSTERS" with [f] is inferred from the G transcription *OF HOLSTERS*. For limitations of inferring pronunciations via characters, see Section 4.4.6. Even with E2E-ASR mistranscription, P scored a lower CER than G.

Dataset	GM	PM
In_LJ	13.9 ±3.8	12.1 ±3
Out_LJ	32.7 ±4.4	24.4 ±3.2

Table 4.7: Results of *In-LJ* and *Out-LJ* for GM, PM.

In-LJ			Out-LJ		
Reference	**G**	**GM**	**Reference**	**G**	**GM**
ASTONISHED	ITS FINISHED	ASTONISHED	COMMEMORATIONAL	COMISERATION	COMMEMORATIONAL
EDITORS	WHAT IT IS	EDITORS	DEMAGNETISED	TO MANETIST	DEMAGNETISED
INTESTINES	INTESTINGS	INTESTINES	HYPNOTISABLE	HICKS NOTICEABLE	HYPNOTIZABLE
INADEQUACY	INECCLES	INADEQUACY	ISLAMICIZE	AS LEMON HES	ISLAM AS IT IS
MAGNIFYING	MAY NOT FIND	MAGNIFYING	OPTIMISES	UP TO MISSUS	UP TO MYSES
REINFORCE	RAIN FORCE	REINFORCE	SORROWFULNESS	SO RUFFLEDNESS	SORROWFULNESS
REORGANIZING	WHERE IT IS IN	REORGANIZING	UNNAMEABLE	UNANIMABLE	UNNAMABLE

Table 4.8: Words in *In-LJ* and *Out-LJ* that have higher CER for G than GM. In both sets, GM pronounces words of multiple morphemes with higher intelligibility. Since *Out-LJ* contains more multi-morpheme words than *In-LJ*, the CER for *Out-LJ* is lower than it is for G. Some pronunciations can be inferred from the E2E-ASR transcription such as incorrect stress/pronunciation from G for "HYPNOTISABLE" (***NOTI**CEABLE*) and "ISLAMICIZE" (*LEMON **HES***) and "SORROWFULNESS" (*SO **RUFF**LEDNESS*). Despite differences in transcription normalisation (e.g. "UNNAMEABLE" and *UNNAMABLE*, the transcriptions of GM were more accurate than G.

may not be picked up in general naturalness tests as with surrounding context of other words and at a sufficient speed the effect may be masked (as argued in [177]). Pronunciation errors were still observed where morpheme boundaries did not disambiguate for instance "SHEPHERDING" was transcribed as *SHEFFORDING* and "DUMBLY" as *DOUBLY* since "SHEPHERD" and "DUMB" were the morphemes and were not decomposed around confusing letters (*PH* and *MB* respectively).

Between GM and PM, no significant difference is recorded with the *In-LJ* set. For the *Out-LJ* set, P performed at a similar CER to GM. GM scored a significantly higher CER than PM but both were significantly lower than system G in Table 4.5. PM scored notably lower CERs than P but the difference was not statistically significant.

4.4.6 Can E2E-ASR detect Correct Pronunciation?

The examples presented in Tables 4.3, 4.4 and 4.6 demonstrate how implicit pronunciation modelling unsuccessfully generalises to words where the pronunciation is dif-

ficult to predict from context. Examples are also shown where with information from the pronunciation lexicon pronunciations can sound more adequate.

However, as mentioned in Section 4.4.4.2, the token words in carrier sentences present a contrived context for E2E-ASR transcription. Token words pronounced in citation form may not represent a deployed use-case since citation form may not be the only viable pronunciation. For instance, a pronunciation that transcribes "REINFORCE" as *RAINFORCE* is acceptable.

E2E-ASR is ultimately too limited in capturing pronunciation variation (e.g. different vowel qualities across accents of English). Speech in context may drop sounds. In [28] the example is given that the words *I don't know* could have multiple realisations which are each valid and intelligible in context:

[aɪ dəʊnt nəʊ]

[aɪ dʊnəʊ]

[dʊnə]

[ə̃ə̃ə̃]

Since E2E-ASR creates a text transcription, the nuance in phonetic realisation is not explicitly detailed and acceptable elision may lead to transcription errors of valid pronunciations.

Due to these limitations combined with issues resulting from text-normalisation (e.g. homophonic transcription) E2E-ASR does not provide a reliable transcription in absolute terms. At best, E2E-ASR can detect relatively better pronunciations when testing on a large scale, but even then E2E-ASR is liable to mistranscription for words of difficult G2P relations (as shown in the examples of Table 4.3).

4.4.7 What is Correct Pronunciation?

Implausible pronunciations by TTS systems with text-input have been exemplified in multiple tables across preceding chapters - including pronunciations which are difficult to predict from the immediate surrounding context alone (such as foreign words). The term "pronunciation correction" could be interpreted as adopting a prescriptivist approach to language use. This would be the case if this thesis analysed pronunciations by humans. However, when the term is used here, the aim is to improve implausible pronunciations from TTS systems with text-input. In the next chapter, representation mixing is examined to improve upon implausible pronunciations from TTS systems with text-input.

4.5 Summary

4.5.1 Summary of ASR for Pronunciation Evaluation

1. *What are the qualitative differences between ASR and human-transcription?*

ASR may transcribe TTS babble resulting in misleading accuracy scores without intervention.

2. *What kind of transcription errors does E2E-ASR make?*

Errors were observed relating to text normalisation (see Section 4.3.11). and (near-)homophonic transcription (see section 4.4.4.1).

3. *How fair is the evaluation of pronunciations of difficult G2P words by E2E-ASR?*

While the transcription of natural speech of the difficult G2P words had high accuracy (`N` in Table 4.5), examples in Tables 4.3 and 4.4 show that transcription via text introduces a bias that favours the mispronunciation of difficult G2P words.

4. *Are significant differences observed between input-types when targeted stimuli are used at scale?*

Tables 4.5 and 4.7 show significant improvements with phones and morphological input (`PM`) over text-input alone (`G`). However pronunciation evaluation via text disregards nuances that may matter in real communicative contexts as explained in Sections 4.4.6 and 4.4.7.

4.5.2 Summary Remarks

An analysis of E2E-ASR as an objective measure of TTS intelligibility was presented. E2E-ASR system based on a Transformer model was found to be more reliable to compare intelligibility of SUS stimuli in the Blizzard Challenge than the Challenge's non-native speech experts (ES) and online volunteers (ER). Similar significant groupings were identified by Paid Listeners (*EP*) and *ASR*, and more fine-grained differences were observed when increasing the number of stimuli in *Extra ASR*.

The chapter proceeded with an analysis of E2E-ASR to transcribe words of difficult G2P relations and multiple morphemes. Pronunciations of TTS systems from two sets of words (*In-LJ* and *Out-LJ*) were compared. Transcriptions were unreliable for these

sets due to differences in text normalisation, spelling, (near-) homophonic transcription and a bias towards the transcription of words with difficult G2P relations. However, further examples were presented of mispronunciations from an E2E-TTS trained with text-input (G) and on aggregate this system obtained higher CERs than systems which made use of the pronunciation lexicon (either phones or morphemes). An additional factor in the unreliability of ASR transcription is that word pronunciations are not fixed and can change according to speaker and communicative contexts. In Chapter 5, improvements in pronunciation modelling are assessed with as small a lexicon as possible under representation mixing.

Chapter 5

Representation Mixing for Pronunciation Correction

Mispronunciations in E2E-TTS can be corrected using representation mixing where text-input is substituted for phone-input when necessary [111]. Since the pronunciation lexicon presents an obstacle to fully E2E-TTS, it would be desirable to minimise the number of word entries in the lexicon needed for this purpose. In this chapter, experiments are conducted aiming to minimise the size of the lexicon required for pronunciation correction when using LJ Speech for training Tacotron in English. A small-scale subjective evaluation is contrasted to a large scale objective evaluation using the E2E-ASR model from the previous chapter.

5.1 Motivation

Representation mixing involves training on a mixture of text- and phone-input, with each input word represented either as graphemes or phones. With the option of using phone-input, it becomes possible to control pronunciations at test time without the need for a complete lexicon of all words in the training data. To my knowledge, the first proposal of representation mixing for E2E-TTS was in the Deep Voice 3 paper [111]. The approach was also described in [23]. However, previous work on representation-mixing had not empirically studied the robustness of pronunciation control or correction. For a functional phone corrector, a certain amount of training data must be labelled with phones. However, building a high-quality pronunciation lexicon can be costly. The rational behind the following experiments was to discover how much pronunciation correction was possible with lexica of different sizes.

5.2 Research Questions

1. How much pronunciation correction is possible with as small a lexicon as possible?

2. Is pronunciation correction possible with only single occurrences of phone-labels during training?

5.3 Representation Mixing

5.4 Method

We[1] closely followed the representation-mixing training approach detailed in [23]. During training, a word in text-input can be replaced by its phone string. This occurs at a fixed mixture probability of p_{mix} only for words in the lexicon being used. We simulated lexica of different sizes and word types as subsets of Unisyn. Representation-mixing was incorporated into the Tacotron [7] implementation used in Chapter 3 [202]. The Tacotron model predicts mel-spectrogram frames, from which we use WaveRNN [113] (a single model trained on the LJ Speech corpus is used in all models) to generate waveforms. The same default training schedule was used as before and each model was trained on a single Nvidia GTX 1080.

We analysed 3 factors in pronunciation correction: the number of word types that are phone-labelled in the training data, the criteria according to which these word types are selected (randomly, or by frequency), and whether coverage-based selection algorithms can reduce the amount of phone-labelling needed. We obtained phone sequences from the Unilex GAM pronunciation lexicon, for its wide phone coverage (167,000 entries), consistency in phone labelling and additional linguistic metadata (stress and syllable boundary information) which we used in our experiments. The phoneset consisted of 56 phones (55 of which were found within LJ Speech). The *x* phone as in *loch* was missing. All models were trained with the full LJ Speech dataset of 24 hours, instead of the IV subset used in Sections 2.6 and 3.3.5.

[1]This work was co-authored with Jason Fong who wrote the code for the representation mixing and trained the Tacotron models. He also designed and wrote the code for the algorithm behind the `phone`, `bigram` and `trigram` models. The listening test was conducted together. After our initial experiment, additional TTS stimuli were generated and then evaluated using the E2E-ASR model from the previous chapter.

5.4.1 Simulated Lexica

To investigate the limits of how well representation-mixing can perform pronunciation correction, we simulated a range of pronunciation lexica differing in number and choice of word types. The word types contained in a given lexicon were phonetised according to the mixture probability, p_{mix}. Three reference lexica were designed as follows:

- `grapheme-only`: an empty lexicon; training a representation-mixing model with this lexicon was equivalent to training with grapheme- or text- only input. This model was used to determine mispronunciations in Tacotron.
- `oracle-14`: contained 14 word types. This was the smallest possible lexicon that covered all 55 phones that occur in LJ Speech at least once. We used this lexicon to discover whether minimal phone coverage was sufficient to enable pronunciation control. Note, it is named with `oracle` since phone labels of the complete LJ dataset were required to build this reference.
- `full-13049`: all 13,049 word types that co-occur in LJ Speech and Unilex. Note this was not equivalent to a phone-based model (e.g. `P` in previous chapters). Recall each word token during training is input as either graphemes or phones according to p_{mix}, so long as the word token belongs to the set of word types in the lexicon.

We additionally devised 5 word type selection methods that each lead to a lexicon of n entries. We compared models trained with these lexica to determine the most effective size and contents for a resource-limited lexicon. For each of the word type selection methods, we varied the number of types `n`, in the lexicon. We selected the following values for `n`: 500, 2000, 4000 and 6000. Each word type selection method was thus tested to determine an effective lexicon size for pronunciation correction. The 5 word type selection methods were:

- `rand-n`: randomly selected n word types. This selection method functioned as a baseline in terms of a word selection method.
- `freq-n`: selected the top n most frequently occurring word types in LJ Speech. This selection method investigated the effect of choosing types to cover the most tokens during training.

- `phone-n`, `bigram-n` and `trigram-n`: these attempted to achieve wide phone coverage during training. `phone-n` greedily selected n frequently occurring word types while also trying to achieve a wide phone coverage. It employed the full lexicon to obtain phonetic knowledge. In `bigram-n` and `trigram-n`, wider contexts from surrounding graphemes are used to replace the oracle knowledge from the lexicon. For details of this algorithm, see our paper [265].

We trained one model for each reference lexicon – `grapheme-only`, `oracle-14`, `full-13049` – and one model for each combination of word type selection method and value of n in $\{500, 2000, 4000, 6000\}$, for a total of 23 models.

In our original experiment, 5 supplemental models were trained using the `full-13049` lexicon to answer further questions not related to lexica size or composition:

- `mixprob-up` and `mixprob-down`: linearly varied p_{mix} so that the probability a word was phonetised during training depended on the frequency rank of the word. `mixprob-up` phonetised the most common word in LJ Speech with $p_{mix} = 0.5$, and the least common word with $p_{mix} = 0.9$. These values were swapped for `mixprob-down`. These models investigated whether phonetising the most or least frequent words more often would benefit pronunciation correction.

- `syllable`, `stress`, and `stress-syllable`: these models additionally include a word type's stress and/or syllable information when it was phonetised during training. Since analyses in previous chapters observed mispronunciations relating to subword units and stress, these models investigated whether additional linguistic markup would benefit pronunciation correction. Note, syllables and stress markup was extracted from Unilex between phones not graphemes. For instance, the entry for *speechless* was `/s p ii ch 1 | lw @ s 0/`, where digits encoded syllable stress and the `|` symbol represented syllable boundaries.

Table 5.1 shows how many tokens in LJ Speech were covered by each lexicon (expressed as a percentage of the tokens covered by `full-13049`). Each of these tokens were randomly phonetised during training.

5.4.2 Test Sets

In our original experiment, we created 3 test sets of words. Each word was placed in the carrier sentence "*Now we will say ... again.*" as previous targeted stimuli in this thesis. The test sets are described below

n	500	2000	4000	6000
rand	3%	19%	43%	56%
freq	69%	86%	93%	96%
bigram	55%	75%	85%	90%
trigram	44%	49%	65%	72%
phone	66%	81%	90%	94%

Table 5.1: Number of word tokens in LJ Speech covered by each resource-limited lexicon expressed as a percentage of the 223179 tokens covered by `full-13049`. Additionally: `oracle-14` covers 41 tokens (0.018% of `full-13049`).

- $In - LJ_{small}$: 50 words that occurred in LJ Speech but were mispronounced by the `grapheme-only` model. Despite being in the training data, they were mispronounced. This set investigated pronunciation correction for word types seen (as either graphemes or phones) during training.

- $Out - LJ_{small}$: 50 words that did not occur in LJ Speech, and were mispronounced by the `grapheme-only` model. These represented the key challenge of generalising to words without spoken examples in training.

- $Cor - LJ_{small}$: 50 words that occurred in LJ Speech and were pronounced correctly by the `grapheme-only` model. This test set checked that representation-mixed training preserved correct output from the grapheme-only model. All models except `oracle-14` scored 47/50 or above. This suggested representation-mixing did not negatively impact pronunciations that a `grapheme-only` model already pronounced correctly, although the results suggest that a lexicon of at least 2000 word types may be required in practice (see below).

Judgements of pronunciation correctness require careful listening so we used 2 expert listeners in our original experiment. The listeners judged whether the pronunciation was correct for each sample in every set. The samples were presented in random order. The listeners were provided with the intended pronunciation for each stimulus. In cases of disagreement, they discussed and re-listened to reach an agreement.

After this experiment, samples were subsequently generated from two sets used in Chapter 4: *In-LJ* and *Out-LJ*. These sets used the same carrier sentence. `In_LJ` contained 3,000 words randomly selected from LJ Speech. Whereas $In - LJ_{small}$ contained seen words that `grapheme-only` mispronounced, the words in *In-LJ* were randomly

	In-LJ small (/50)	OO-LJ small (/50)	In-LJ (Acc%)	OO-LJ (Acc%)
oracle-14	0	0	14.1 ±.8	14.1 ±.7
full_13049	47	38	86.7 ±0.9	70.1 ±2.0
syllable	47	48	89.3 ±.7	79.6 ±1.0
stress	31	33	66.5 ±2.4	52.6 ±2.1
stress-syllable	46	45	84.3 ±1.5	73.9 ±1.4

Table 5.2: Results from listening test and ASR. ± indicates 2 standard deviations of the 95% confidence interval.

selected from LJ Speech without the pre-requisite of a mispronunciation. *Out-LJ* contained 3,000 words of inaccurate G2P according to the `LJ Token` G2P model in Chapter 2. For *In-LJ* and *Out-LJ* the CER method (described in Section 4.3.11) was used to calculate the accuracy of the target words. Accuracy was presented to ease interpretation between Figures 5.1 and 5.2. While this model was shown to be unreliable for the transcription of specific words, at scale differences in overall speech quality may still be observed as was shown in Section 4.4.5.

5.5 Results

5.5.1 Syllable and Stress Results

The results[2] for the models using the reference lexica and linguistic metadata are presented in Table 5.2. The minimal phone coverage model `oracle-14` scored 0 on $In-LJ_{small}$ and $Out-LJ_{small}$. It also scored the lowest accuracy on *In-LJ* and *Out-LJ*. Evidently, the coverage was insufficient to perform any pronunciation correction - the output was unintelligible. Thus, pronunciation correction with only 14 word types phonetised during training fails.

The `grapheme-only` model scored 0/50 for $In-LJ_{small}$ and $Out-LJ_{small}$ and 76.9%±2.3 and 53.1%±1.7 accuracy for *In-LJ* and *Out-LJ*. The score for *Out-LJ* was similar to the `G` model from Chapter 4 but the CER for *In-LJ* was significantly lower. Performance was expected to be better for these models since `grapheme-only` was

[2]Samples are available at `https://jonojace.github.io/IS20-repmixing-limits`

trained on approximately 25% more data than `G`. The scores for both sets were significantly lower than `P` from Chapter 4.

The full-phone coverage model (`full-13049`) performed with 47/50 for $In-LJ_{small}$, 38/50 for $Out-LJ_{small}$ and with 86.7% and 70.1% accuracy for *In-LJ* and *Out-LJ* respectively. `full-13049` can be treated as a baseline for pronunciation correction. Due to lack of context, it is unrealistic to expect 50/50 or 100% accuracy with carrier sentences.

The accuracy for *In-LJ* was higher (but not significantly higher) than `P` from Chapter 4, trained solely on phone-input. This may be caused by increased dataset size. Another explanation could be that the combination of grapheme- and phone- input improve overall pronunciation modelling. Recent work such as [149] used a combination of graphemes and phones in a BERT encoder which they argue improves prosody and pronunciation of Tacotron.

`Syllable` matched `full-13049` on $In-LJ_{small}$ and outperformed `full-13049` on the other 3 test sets. This suggests pronunciation modelling with phone-input can be further improved with syllable boundaries. The increase in accuracy was relatively higher (9.5% rather than 2.6%) for *Out-LJ* than for *In-LJ*, although following the analysis of Section 4.4 it is unclear from the CER metric alone to what this difference may be attributed at scale (overall improved speech quality with fewer skips/deletions or better G2P generalisation?). Listening to samples, the speech sounds more natural. Future work could explore whether unsupervised methods for subword decomposition improve implicit pronunciation modelling under representation mixing: do units need to be linguistically symbolic (e.g. syllables, morphemes) or can units be based on frequency of character contexts (e.g. BPE)?

`Stress` scored beneath `full-13049` on all test sets. When stress markers were used in combination with syllable boundaries (`stress-syllable`), the performance was lower than when using syllablic boundaries without stress (the `syllable` model). Stress markers do not improve pronunciation modelling or correction under representation mixing.

Results for `mixprob-up` ($In-LJ_{small}$:47/50, $Out-LJ_{small}$: 42/50, `In-LJ`: 87.9%$\pm$2.1, `Out-LJ`: 75.2%$\pm$1.6) demonstrate that phonetising lower frequency words with a higher p_{mix} during training slightly improved pronunciation control for *Out-LJ* words, compared to the uniform $p_{mix} = 0.5$ across all word types used in the `full-13049` results above. Results for `mixprob-down` ($In-LJ_{small}$: 48/50, $Out-LJ_{small}$: 39/50, `In-LJ`: 86.1%$\pm$0.8, `Out-LJ`: 71.6%$\pm$1.8) were very similar to `full-13049`, indicat-

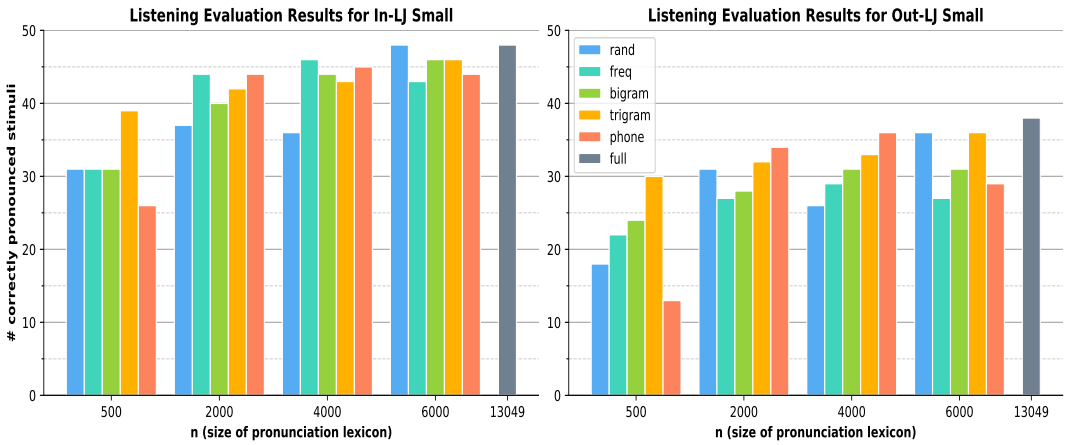

Figure 5.1: Listening test results of models trained using resource-limited lexica generated by the word type selection methods.

ing phonetising lower rather than higher frequency words with a higher p_{mix} during training is more beneficial.

5.5.2 Word Type Selection Results

5.5.2.1 Listening Test Results

Figure 5.1 visualises the listening test results when word selection method types were scored by two expert listeners. The left of Figure 5.1 shows scores (out of 50) for *In-LJ Small*. For the test sets in Figure 5.1, `graphemes-only` scored 0.

The representation mixing models scored higher accuracy on $In-LJ_{small}$ than on $Out-LJ_{small}$, as expected. Overall, pronunciation correction was higher the larger the n word types phonetised, particularly when n was increased from 500 to 2000. `Trigram-500` outperformed the other word type selection methods with $n = 500$.

For $Out-LJ_{small}$, representation mixing with `full-13049` achieved 38/50 as a platform. This score was only improved upon when using syllable boundaries (48/50). As mentioned above, the use of syllable boundaries could be investigated further to find out whether subword units need to be linguistically symbolic to improve pronunciations of difficult words (for example to disambiguate '*th*' in words like *pothole* and *goatherd*).

The above results indicate that pronunciation correction is possible from a lexicon of potentially only 500 words. However, what insights did the ASR evaluation bring?

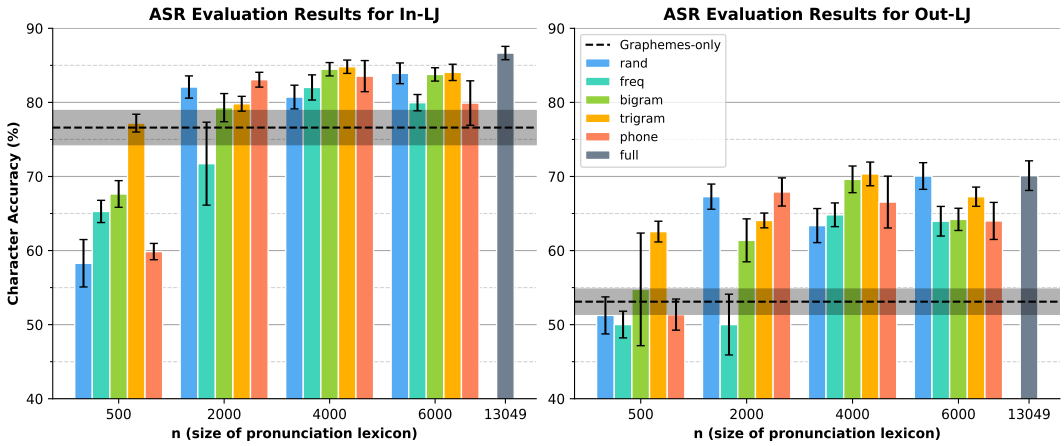

Figure 5.2: ASR results of models trained using resource-limited lexica generated by the word type selection methods. The accuracy and confidence interval of `graphemes-only` is shown by the dotted lines and shaded areas. Representation mixing with $n > 4000$ consistently obtains higher character accuracy for *Out-LJ* than `graphemes-only`. `Trigram-500` obtained the highest accuracy amongst the methods with n=500. Some examples of corrected pronunciations are shown in Table 5.3. The accuracy of $n = 500$ for *In-LJ* are lower than `graphemes-only` indicating that representation mixing with too small a lexicon worsens intelligibility.

5.5.2.2 ASR Evaluation Results

Figure 5.2 shows the transcription accuracy of word type selection methods according to size n of lexicon. The accuracy and confidence interval of `graphemes-only` are shown by the shaded dotted lines in each subfigure. Note the *In-LJ* set for the ASR evaluation did not select words for G2P difficulty as in $In-LJ_{small}$. The *In-LJ* set contained words randomly selected from the training data of LJ speech.

On *In-LJ* the models with $n = 500$ scored lower accuracy than `graphemes-only` except for `trigram-500`. While the small test with $Cor-LJ_{small}$ did not show extensive pronunciation errors, the models have lower accuracy when transcribing random words (from *In-LJ*) using ASR. The indication is that with only 500 word types phonetised, intelligibility when using phones is below using `graphemes-only`. This means that for most word selection methods, $n = 500$ is detrimental to pronunciation modelling. However, all models with $n \geq 2000$ matched or exceeded `graphemes-only` in accuracy on *In-LJ* (except for `Freq-2000`). However, the gap between `graphemes-only` and the models with $n \geq 2000$ are small in some cases insiginificant (e.g. `bigram-2000`, `Trigram-2000`, `phone-6000`). These results suggest that there may be some small im-

Reference	graphemes-only Hypothesis	graphemes-only Acc	Trigram-500 Hypothesis	Trigram-500 Acc
BLOODS	BLOWS	60	BLOODS	100
CAGYNESS	TUGGING US	33.3	CASHINESS	66.7
CHICAGOS	CHUCKED US	44	CHICAGOS	100
DOOGLEBUG	DONT LOOK AT IT	25	DO THE BOAT	33.3
HEINRICH	HE RICH	66.7	HEINRICH	100
LIECHTENSTEIN	LAKE AND SANE	0	LIECHTENSTEIN	100
LOGANBERRY	LA GAMBORE	44.4	LOW AND BEGGARY	61.5
MEGALOMANIAC	MC GILL AMMUNIE	38.5	MY DOLOMANIA	63.6
MOVEABLE	MOVE YOU WILL	45.5	MOVABLE	85.7
OESOPHAGUSES	WAS THE PAGES	27.3	ASSOPHODACES	50
PHILOSOPHIZERS	PHYLLOSOPHYSORS	66.7	PHILOSOPHIZERS	100
PIGEONHOLE	PUDGEN HALL	50	PIGEON HOLE	100
PORTSMOUTH	POURED A MOUTH	66.7	PORTSMOUTH	100
SIMONS	SUMMONS	71.4	SIMONS	100
SPONGED	FOND	0	SPONGE	83.3
THAILAND	NOTHING	14.3	TALL AND	71.4
TOOTHACHE	TO FETCH	28.6	TO THEE	33.3
WALES	WAS	33.3	WAILS	60
WISEST	WIZARD	33.3	WISEST	100

Table 5.3: E2E-ASR transcription of `graphemes-only` and `Trigram-500` representtion mixing model. Some mispronunciations can be observed in the E2E-ASR transcription such as PORTSMOUTH with the word *MOUTH* [maʊθ]. Note: whitespace was removed before calculating accuracy. Accompanying audio samples are available at `https://homepages.inf.ed.ac.uk/s1649890/chap5/`

provements with representation mixing for random words, but the overall effect size is small.

What about pronunciation correction for the *Out-LJ* set? Recall from Chapter 4 when transcribing natural speech, the ASR accuracy scores were: 7.7%±2.9 and 13.6%±6.3 for *In-LJ* and *Out-LJ* respectively, which was a decrease in accuracy by 5.9% when transcribing more difficult words. As with other models in Section 4.4.5, the accuracy scores for *Out-LJ* were lower than for *In-LJ*. However, accuracy was reduced more for `graphemes-only` than for `Trigram-500` and all models with $n \geq 2000$ (except for `Freq-2000`). The larger gap between `graphemes-only` and these models for *Out-LJ* show improved pronunciation modelling with representation mixing. Despite the imperfections in using E2E-ASR transcription to evaluate pronunciation modelling, the gap between `graphemes-only` and the models with $n \geq 2000$ overall is larger when transcribing words of difficult G2P relations. E2E-ASR is still detecting larger differences in quality for these sets of words when using representation mixing. Some examples of ASR transcription of words from *Out-LJ* (with difficult G2P rela-

tions) are presented in Table 5.3. The reader is encouraged to listen to samples via the link in the Table's caption.

Table 5.3 shows that the implicit G2P model of `graphemes-only` incorrectly generalises pronunciations of difficult words. Table 5.3 also shows that better pronunciations from `Trigram-500` were transcribed with higher accuracy.

One main takeaway from this analysis is that E2E-ASR identifies larger gaps between `graphemes-only` and the word type selection methods for difficult G2P words (*Out-LJ)* than for random words (*In-LJ*). The other main takeaway is that pronunciation correction is possible with $n = 500$ if the word types are selected to obtain a wide phonetic coverage as in the `trigram` method. Overall, representation mixing with a pronunciation lexicon is beneficial for E2E-TTS, but a large lexicon may not necessarily be required to correct pronunciations.

5.6 Summary

5.6.1 Summary of Pronunciation Correction Experiments

1. *How much pronunciation correction is possible with as small a lexicon as possible?*

With judicious selection of word types (exemplified by `trigram-500`), a high degree of pronunciation correction was possible with only 500 words in a lexicon. However, with other selection methods (e.g. random word type selection) a lexicon of at least 4,000 words more reliably rendered pronunciation correction.

2. *Is pronunciation correction possible with only single occurrences of phone-labels during training?*

No. Single occurrences of phone labels (in the `oracle-14` model) rendered unintelligible speech with no pronunciation correction.

5.6.2 Summary Remarks

Pronunciation correction using representation mixing was analysed in a small-scale expert-based listening test and in a large-scale evaluation using ASR. A lexicon of 500 words in `trigram-500` was shown to correct mispronunciations made by a model trained only on text-input `graphemes-only`. Pronunciation correction was improved

the most by incorporating syllable boundaries with phones during training and testing. Despite the imperfections with the E2E-ASR model to evaluate difficult pronunciation phenomena, significant improvements in accuracy were still observed over a `graphemes-only` model with representation mixing using lexica of $n \geq 2000$. The next chapter presents my thesis, some concluding remarks and directions for future work.

Chapter 6

Conclusions

In the foregoing chapters, I have analysed pronunciation modelling in DC-TTS and Tacotron to assess the viability of E2E-TTS without the need for a pronunciation lexicon. In Chapter 2 initial attempts to evaluate implicit pronunciation modelling via G2P and a MUSHRA were conducted. Simulation of the implicit pronunciation model via G2P revealed words that were subsequently mispronounced by DC-TTS (and Tacotron). The chapter concluded that stimuli containing G2P error words would be suitable to evaluate the need for a pronunciation lexicon in English.

In Chapter 3, further G2P experiments were conducted finding an improvement to the implicit pronunciation model with gold-standard morphological labels at input. Experiments were also conducted in French concluding that other aspects of the traditional TTS front-end (post-lexical rules) are still beneficial for the control of linguistic phenomena such as liaison. The chapter called for a more objective and scalable evaluation method for pronunciation in speech to assess the value of the lexicon (and the broader front-end) for Tacotron.

In Chapter 4 ASR was assessed for pronunciation evaluation. Across multiple years of the Blizzard Challenge, ASR transcriptions were found to be more reliable than untrained listeners. However, for the specific task of pronunciation evaluation, text transcriptions were shown to mask pronunciation errors made by Tacotron. Despite this bias, when Tacotron used phone-input on a large set of G2P error words ASR transcriptions were still more accurate than when Tacotron used text-input.

In Chapter 5, the low-resource solution of representation mixing for pronunciation correction was investigated. Both a small-scale expert listening test and a large-scale ASR evaluation showed representation mixing can successfully be used to correct pronunciations with a lexicon of only 4,000 words, and potentially with only 500 words

if these are selected judiciously for phonetic coverage. The takeaway messages from these chapters are presented as my thesis below.

6.1 Thesis

- Pronunciation control is desirable in certain deployable TTS applications to ensure correct pronunciation.
- The pronunciation of some words cannot be predicted by generalising from surrounding character contexts alone. To accurately guide the correct pronunciation of these words requires prior knowledge as represented in a lexicon.
- E2E-ASR transcription can be more reliable than transcription by unreliable judgements from untrained listeners to approximately evaluate the intelligibility of TTS systems. However, E2E-TTS transcription can be unreliable for words of difficult G2P relations.

6.2 Current answers to anticipated questions from the reader

Certain questions will have occurred to the reader during the course of the preceding chapters. In this section, the most obvious questions are addressed leading to directions for future work.

6.2.1 Is correct pronunciation important?

As described in Sections 4.4.6 and 4.4.7, what is understood to be an adequate pronunciation may depend on a communicative context. This may lead the reader to question the utility of gold-standard phone labels: why should slight mispronunciations be considered unacceptable? Does it really matter if *karate* is pronounced [kəreɪt] or pothole is pronounced with a [ð] sound?

The answer depends on the researcher's resource allocation decisions and the frequency with which a language exhibits pronunciations not reliably predictable from character context. In a high resource language such as English in a deployed use-case such as smartphone voice applications, correct pronunciation of words with difficult G2P relations may be very important (as also argued in [111]).

6.2.2 Why use the pronunciation lexicon when the idea of E2E is not to use manually created resources?

Work on E2E-TTS can employ a G2P model instead of a pronunciation lexicon to obtain phone strings. However, this approach may still provide an unsatisfactory pronunciation guide. This is important to point out when works such as [5], [173] question the value of using phones in E2E-TTS. The value of gold-standard phone labels goes hand-in-hand with the issue of one's available resources and the importance one allocates to control over pronunciation.

A related question may ask that since word pronunciations are not fixed and may vary on communicative context, why should one have to specify the pronunciation? The answer depends on the importance one attaches to the pronunciation of some words.

On the value of the pronunciation lexicon for words with difficult G2P relations, some recent work on E2E-ASR should also be mentioned. Despite non-significant differences in performance in E2E-ASR with text- or phone-input, there are proposals to improve transcription of difficult G2P words in E2E systems [257], [266]. However, these offer improvements but do not provide the assurance of a lexicon.

6.2.3 Does more data improve pronunciations of E2E-TTS?

This point requires objective clarification, perhaps through a systematic study as conducted for ASR in [18]. The results of a CBHL encoder [173] with 24 hours of LJ and a CBHG module using 39 hours of training data in [4] were shown to have similar performance between text- and phone-input. However in the results with a CBHG module in Chapter 3, a MUSHRA (which assesses the same test stimulus directly alongside all systems under test - unlike a MOS test) showed a difference in performance when trained on only 18 hours of data. The dataset may interact with other features too (the language, the audio quality, single or multi-speaker). Given these different but difficult-to-compare results, future work suggests a systematic comparison that takes into account potentially interactive factors too (see Section 6.3 below).

6.2.4 Is a pronunciation lexicon needed in languages other than English?

The relative benefit of a lexicon may lie in the relationship between G2P relations (or LTS rules) and the number of words in a lexicon, as shown in Figure 2.2. Comparing the relative G2P performance of pre-trained E2E-TTS text encoders systematically in different languages (with a variety of datasets sizes) would be an interesting path for future work (see Section 6.3).

Beyond the pronunciation lexicon, some manually written resources may provide better assurances than predictions based on sequence modelling such as with text normalisation, pitch accent types in Japanese, or semantic rules for polyphone disambiguation in Chinese.

Another related question is how TTS systems should pronounce foreign names in a language. Is nativisation required (where foreign names may be pronounced with the phonotactics of their language of origin [103])? It is unclear how well nativisation works across multiple languages in E2E-TTS.

6.2.5 Why are text- and phone-input regarded as equivalent?

Why do researchers adopt the view that there is no difference between text- and phone-input in E2E-TTS? Firstly, the adoption of S2S acoustic models allowed for improved contextual learning from character-input.

Text-encoders were shown to learn the context of input characters, for example in [5], [8]. Furthermore, in [3] the authors concluded that with a *learned text encoding* in DC-TTS, explicit context features previously used in SPSS were effectively redundant. The implication of this finding it that S2S acoustic models with characters improves contextual pronunciation modelling. Since some ambiguous G2P relations may be learnt by surrounding character contexts given enough data and a large enough model parameter size, so the thinking goes, the gap in performance between text- and phone-input close. This has been shown to an extent empirically with MOS test scores in [4], [173] with CBHL/G encoders (see Section 2.6.4.2)

However, some pronunciations are not learnable from the context contained within a single text-audio pair alone. For instance, it is difficult to generalise pronunciations to words of foreign origin which do not follow the typical G2P patterns of the language's orthography. Examples are provided throughout the preceding chapters in Tables 2.3, 3.3, 3.4, 3.5, 3.6, 4.3, 4.6, and 5.3. To illustrate this point here, let us take one further

example. The G2P model prediction (translated into the IPA) for *karate* from the `LJ Types` G2P model was [kəreɪt]. More knowledge than the surrounding character context would inform the model that it is a word of Japanese origin with different G2P relations. Knowledge about the word's pronunciation simply has to be known.

The second reason it is argued text-input is no different from phone-input is because standard listening test data does not adequately test for differences between the input-representations. Since the claim of "*difficult to distinguish from human speech*" has been made [7, p.1], the specific key competence of pronunciation is often disregarded as an issue amongst broader tasks such as improving intonation or prosody for expressive TTS. Evaluations on a held-out set often tests a general but vague notion of naturalness whereas stimuli specifically selected according to criteria can offer further insight. In this thesis, words selected for ambiguous G2P relations have shown differences between text- and gold-standard phone-input.

A related point was made in [177] that large-scale evaluations can mask small but potentially important differences in TTS systems. In their approach, they proposed a selection of stimuli according to acoustic dis-similarity between TTS systems. This approach could be incorporated in the future to evaluate pronunciations in TTS systems. The point of contention here is: if an error is masked in a large scale evaluation, then is it unimportant? In certain deployment cases (for instance smartphone applications) the pronunciation of certain words such as names may be very important for the user.

The third reason is researchers may be using phone labels from a G2P model. A G2P model may equally have generalisation issues for words that are not easily predictable from character context. In Chapter 2, G2P models trained on words from E2E-TTS datasets had higher error rates than a G2P model trained on the Combilex lexicon. Even with improved S2S G2P generalisation (be it from more data or improved deep learning model architectures), the pronunciation for some words cannot be predicted from generalised G2P relations based on character contexts alone.

The fourth reason text- and phone-input are treated with equivalence is when English is cast aside from other languages for its irregular orthography. However, predicting pronunciations from character context in an input-string can pose issues in other languages too. For instance, the incorrect insertion of disallowed *liaison* in French (see Table 3.6) or the pronunciation of compound words in Kiswahili - see Section 3.4.2). Some languages have a large set of input graphemes (e.g. logographies) which require a conversion to a phonetic alphabet - such as Mandarin or Japanese.

6.2.6 How does the performance of implicit G2P modelling differ across E2E-TTS architectures?

Another salient question from the reader may be that of the interplay between architecture choice and learning from text. Besides the comparison of the CBHL and CNN encoders in [173], how are pronunciations handled in more novel architectures that potentially exploit implicit semantic context such as BERT? In [171] a machine learning approach to homograph disambiguation used part-of-speech (POS) features. How successful is homograph disambiguation in E2E-TTS? Does the use of a BERT network [149] exploit implicit semantic information to learn the difference between (*sea-)bass* and *(musical-)bass*?

There are more interesting questions waiting to be answered. What is the effect of dataset size on the performance between text encoders trained with text- and phone-input? How beneficial are gold-standard phone labels compared with predicted labels from an external G2P model? How are pronunciations modelled in context in multi-speaker synthesis across accents? In EATS [20], phone-input was observed to be significantly better but which dialectal variants cannot be learnt from context alone? As mentioned in Section 2.6.4.2, a systematic review of text- and phone-input across different E2E-TTS architectures would be insightful. However, ultimately certain words cannot be modelled from character context alone.

6.2.7 How could latent pronunciation knowledge in E2E-TTS models be made interpretable?

The reader may also question the nature of the latent knowledge inextricably entwined in the parameters of neural networks. How can this knowledge be made interpretable and differences in input representations be contrasted in TTS? Interpretability of neural networks ultimately seeks an explicit, contradictory-free set of explanations for combined latent decision making by the set of all weights. Fundamentally, a full explanation defeats the object of employing such an approach in the first place. Only vague insights can be learnt, the complete *rule-based* answer will not be revealed.

Notwithstanding this limitation, further work on understanding how language is implicitly learnt would be interesting future work - in particular to study the effect of subword decomposition on neural sequence modelling in more detail. Do subword units for TTS need to be linguistically symbolic (syllables/ morphemes) or can they be

unsupervised (BPE or Morfessor units)? Is there an interaction here with the language being processed?

By adopting the taken approach in [8], E2E-TTS can be treated as classifiers of morphological boundaries to understand how much morphological knowledge is learnt. More broadly, the approach could also be applied to different architectures. Phones could be classified from text-input and the pronunciations for words with difficult word pronunciations could be evaluated in this way similarly to the implicit G2P models of Chapter 2.

6.2.8 What are future research directions for TTS evaluation with ASR?

In practise, ASR evaluation via APIs (as used in [21]) will be more robust than the E2E-ASR used in this thesis since they will use a pronunciation lexicon and inverse text normalisation. The findings that ASR is reliable only in a general sense at an aggregate level (and not for particular words) is a warning to researchers that pronunciation analysis at an individual level may still have to be manual.

Nevertheless, a re-evaluation of the Blizzard Challenges in Mandarin would be low-hanging fruit. ESPnet contains an E2E-ASR model for Mandarin and the evaluation data for the Blizzard Challenges in 2019 and 2020 are also freely available online.

6.3 Future work

There remain several interesting unanswered questions about pronunciation modelling in E2E-TTS and E2E-ASR. What I think could be the most insightful directions to better understand the value of expertly created linguistic resources on pronunciation modelling at the current moment are summarised here:

1. To better understand the learning of ambiguous character contexts during training in E2E-TTS, I recommend a systematic evaluation similar to [173] but with varying amounts of training data in multiple languages. The models trained with varying amounts of data, could be treated as classifiers for the task of G2P (for the languages presented in Figure 2.2 or for polyphone disambiguation in Mandarin).

2. How much morphological information is implicitly learnt in E2E-TTS encoders?

A similar experimental setup to the above suggesting but where encoders instead classify morphological boundaries could be an interesting start-point. Do sub-word units need to be linguistically symbolic (e.g. syllables or morphemes) or can unsupervised units (e.g. BPE) be used?

3. How might implicit G2P performance in English vary according to architectures that exploit implicit semantic context such as BERT? In [149], BERT was used in conjunction with phone-input but which is more beneficial for pronunciation modelling: BERT or phone-input?

4. An analysis of input representations in multi-speaker E2E-TTS models would also be interesting. In particular, a contrast between text-, phone- and metaphoneme-input. To what extent do each of these guide pronunciations optimally? How much keyword phonology is learnt in E2E-TTS models?

5. I would recommend a re-evaluation of the Blizzard Challenges in Mandarin for 2019 and 2020 using Automatic Speech Recognition, for comparable analysis to the re-evaluation conducted in English in Chapter 4.

6.4 Some Concluding Remarks

The above directions for future work may bring further insight into the value of expertly created resources in pronunciation modelling, but evidence throughout this thesis shows that a high quality pronunciation lexicon is still more reliable than implicit G2P generalisation from text.

One final point will bring this thesis to a close. Control over pronunciation is important not only for intelligibility but also because an utterance may intentionally express nuances in pronunciation. Speakers may want to highlight how they pronounce words differently across accents, as in the well-known George and Ira Gershwin classic: *Let's call the whole thing off*. Throughout the song, the singers contrast their separate pronunciations of words in two accents (e.g. *either* and *neither*). However in the lyrics, the words may be written identically:

"*You say either and I say either,*
You say neither and I say neither.
Either, either neither, neither

Let's call the whole thing off. - George and Ira Gershwin [267]

It simply has to be known that the first word in each pair is pronounced with an [i] sound and the second word with an [aɪ] sound. So as the song goes: we "*better call the calling off off*" since there is still a place for the pronunciation lexicon.

Bibliography

[1] J. Sotelo *et al.*, "Char2Wav: End-to-end speech synthesis," in *Proc. ICLR*, 2017.

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

[3] O. Watts *et al.*, "Where do the improvements come from in sequence-to-sequence neural TTS?" In *Proc. SSW*, 2019, pp. 217–222.

[4] R. J. Weiss *et al.*, "Wave-Tacotron: Spectrogram-free end-to-end text-to-speech synthesis," in *Proc. ICASSP*, 2021, pp. 5679–5683.

[5] A. Perquin, E. Cooper, and J. Yamagishi, *An investigation of the relation between grapheme embeddings and pronunciation for Tacotron-based systems*, 2021. arXiv: `2010.10694 [cs.CL]`.

[6] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. NIPS*, 2014.

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

[8] K. Mametani, T. Kato, and S. Yamamoto, "Investigating context features hidden in end-to-end TTS," in *Proc. ICASSP*, 2019, pp. 6920–6924.

[9] R. Sproat and N. Jaitly, *RNN approaches to text normalization: A challenge*, 2017. [Online]. Available: `https://arxiv.org/abs/1611.00068`.

[10] A. Black and A. Llitjos, "Unit selection without a phoneme set," in *Proc. IEEE Workshop on Speech Synthesis*, 2002, pp. 207–210.

[11] G. K. Anumanchipalli, K. Prahallad, and A. W. Black, "Significance of early tagged contextual graphemes in grapheme based speech synthesis and recognition systems," in *Proc. ICASP*, 2008, pp. 4645–4648.

[12] O. Watts, J. Yamagishi, and S. King, "Letter-based speech synthesis," in *Proc. SSW*, 2010, pp. 19–26.

[13] R. Kay *et al.*, "Knowledge versus data in TTS: Evaluation of a continuum of synthesis systems," in *Proc. Interspeech*, 2015, pp. 3335–3339.

[14] S. Sitaram *et al.*, "Universal grapheme-based speech synthesis," in *Proc. Interspeech*, 2015, pp. 3360–3364.

[15] K. Ito, *The LJ speech dataset*, Available: `https://keithito.com/LJ-Speech-Dataset/`, 2017.

[16] Y. Yasuda, X. Wang, and J. Yamagishi, "Initial investigation of encoder-decoder end-to-end TTS using marginalization of monotonic hard alignments," in *Proc. SSW*, 2019, pp. 211–216.

[17] T. Sainath *et al.*, "No need for a lexicon? Evaluating the value of the pronunciation lexica in end-to-end models," in *Proc. ICASSP*, 2018, 5859–5863.

[18] M. Zeineldeen *et al.*, *A systematic comparison of grapheme-based vs. phoneme-based label units for encoder-decoder-attention models*, 2021. [Online]. Available: `https://arxiv.org/abs/2005.09336`.

[19] K. Irie *et al.*, "On the choice of modeling unit for sequence-to-sequence speech recognition," in *Proc. Interspeech*, 2019, pp. 3800–3804.

[20] J. Donahue *et al.*, "End-to-end adversarial text-to-speech," in *Proc. ICLR*, 2021.

[21] E. Cooper *et al.*, "Utterance selection for optimizing intelligibility of TTS voices trained on asr data," in *Proc. Interspeech*, 2017, pp. 3971–3975.

[22] E. Cooper and J. Yamagishi, "How do voices from past speech synthesis challenges compare today?" In *submission to SSW*, 2021.

[23] K. Kastner *et al.*, "Representation mixing for TTS synthesis," in *Proc. ICASSP*, 2019, pp. 5906–5910.

[24] J. Wells, *Computer-coding the IPA: A proposed extension of SAMPA*, 1995. [Online]. Available: `https://www.phon.ucl.ac.uk/home/sampa/x-sampa.htm`.

[25] B. Peters, J. Dehdari, and J. van Genabith, "Massively multilingual neural grapheme-to-phoneme conversion," in *Proc. Building Linguistically Generalizable NLP Systems*, 2017, pp. 19–26.

[26] H. Giles, A. Mulac, J. J. Bradac, and P. Johnson, "Speech accommodation theory: The first decade and beyond," *Annals of the International Communication Association*, vol. 10, no. 1, pp. 13–48, 1987.

[27] C. P. Browman and L. Goldstein, "Articulatory phonology: An overview," *Phonetica*, vol. 49, no. 3-4, pp. 155–180, 1992.

[28] R. K. Moore and L. Skidmore, "On the use/misuse of the term 'phoneme'," in *Proc. Interspeech 2019*, 2019, pp. 2340–2344.

[29] L. Loots and T. Niesler, "Data-driven phonetic comparison and conversion between south African, British and American English pronunciations," in *Proc. Interspeech*, 2009, pp. 196–199.

[30] L. Loots *et al.*, "Comparing manually-developed and data-driven rules for P2P learning," in *Proc. PRASA*, 2009, pp. 35–40.

[31] B. Kolluru *et al.*, "Generating multiple-accent pronunciations for TTS using joint sequence model interpolation," in *Proc. Interspeech*, 2014, pp. 1273–1277.

[32] J. C. Wells, *Accents of English*. Cambridge University Press, 1982.

[33] S. Fitt and S. Isard, "Representing the environments for phonological processes in an accent-independent lexicon for synthesis of English," in *Proc. ICSLP*, 1998.

[34] S. Fitt and S. Isard, "Synthesis of regional English using a keyword lexicon," in *Proc. Eurospeech*, 1999.

[35] A. J. Hunt and W Black, Alan, "Unit selection in a concatenative speech synthesis system using a large speech database," *ICASSP*, vol. 1, pp. 373–376, 1996.

[36] H. Zen, *Acoustic modelling for speech synthesis: From HMM to RNN*, Invited talk given at ASRU, 2015.

[37] H. Lu and S. King, "Using bayesian networks to find relevant context features for HMM-based speech synthesis," in *Proc. Interspeech*, 2012, pp. 1143–1146.

[38] R. Dall *et al.*, "Redefining the linguistic context feature set for HMM and DNN TTS through position and parsing," in *Interspeech 2016*, 2016, pp. 2851–2855.

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

[40] 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/CSTR/F0parametrisation/hts_lab_format.pdf`.

[41] A. W. Black, "CLUSTERGEN: A statistical parametric synthesizer using trajectory modeling," in *Proc. Interspeech*, 2006, pp. 1762–1765.

[42] T. Qian *et al.*, "A Python toolkit for universal transliteration," in *Proc. LREC*, 2010, pp. 2897–2991.

[43] Y.-H. Sung *et al.*, "Revisiting graphemes with increasing amounts of data," in *ICASSP*, 2009, pp. 4449–4452.

[44] R. Sproat *et al.*, "Normalization of non-standard words," *Computer Speech Language*, vol. 15, no. 3, pp. 287–333, 2001.

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

[46] S. Yolchuyeva, G. Németh, and B. Gyires-Tóth, "Text normalization with convolutional neural networks," *International Journal of Speech Technology.*, vol. 21, no. 3, 589–600, Sep. 2018.

[47] A. Javaloy and G. García-Mateos, "Text normalization using encoder–decoder networks based on the causal feature extractor," *Applied Sciences*, vol. 10, no. 13, 2020.

[48] H. Zhang *et al.*, "Neural models of text normalization for speech applications," *Computational Linguistics*, vol. 45, no. 2, 293–337, Jun. 2019.

[49] C. Mansfield *et al.*, "Neural text normalization with subword units," in *Proc. NAACL*, Jun. 2019, pp. 190–196.

[50] W. Pei, T. Ge, and B. Chang, "Max-margin tensor neural network for Chinese word segmentation," in *Proc. ACL*, 2014, pp. 293–303.

[51] N. Xue, "Chinese word segmentation as character tagging," in *International Journal of Computational Linguistics*, 2003, pp. 29–48.

[52] G.-T. Liou, Y.-R. Wang, and C.-Y. Chiang, "Text normalization for Mandarin TTS by using keyword information," in *Proc. O-COCOSDA*, 2016, pp. 73–78.

[53] J. Zhang *et al.*, "A hybrid text normalization system using multi-head self-attention for mandarin," in *Proc. ICASSP*, 2020, pp. 6694–6698.

[54] J. Pan *et al.*, "A unified sequence-to-sequence front-end model for Mandarin text-to-speech synthesis," in *Proc. ICASSP*, 2020, pp. 6689–6693.

[55] A. Conkie and A. Finch, "Scalable multilingual frontend for TTS," in *Proc. ICASSP*, 2020, pp. 6684–6688.

[56] R. Sproat and K. Gorman, *A brief summary of the Kaggle text normalization challenge*, 2018. [Online]. Available: `https://medium.com/kaggle-blog/a-brief-summary-of-the-kaggle-text-normalization-challenge-11\\797b7e696f`.

[57] R. Sproat, *Text normalization data for English, Russian and Polish*, 2017. [Online]. Available: `https://www.kaggle.com/richardwilliamsproat/text-normalization-for-english-russian-and-polish`.

[58] S. Tyagi *et al.*, "Proteno: Text normalization with limited data for fast deployment in text to speech systems," in *Proc. NAACL*, 2021, pp. 72–79.

[59] O. Watts *et al.*, "Unsupervised and lightly-supervised learning for rapid construction of TTS systems in multiple languages from 'found' data: evaluation and analysis," in *Proc. SSW*, 2013, pp. 101–106.

[60] S. King, *The simple4all project*, 2014. [Online]. Available: `http://simple4all.org/`.

[61] K. Sodimana *et al.*, "Text Normalization for Bangla, Khmer, Nepali, Javanese, Sinhala and Sundanese text-to-speech systems," in *Proc. SLTU*, 2018, pp. 147–151.

[62] Y. M. Oo *et al.*, "Burmese speech corpus, finite-state text normalization and pronunciation grammars with an application to text-to-speech," in *Proc. LREC*, 2020, pp. 6328–6339.

[63] S. Ritchie *et al.*, "Data-driven parametric text normalization: Rapidly scaling finite-state transduction verbalizers to new languages," in *Proc. SLTU*, 2020, pp. 218–225.

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

[65] W. M. P. Daelemans *et al.*, "Language-independent data-oriented grapheme-to-phoneme conversion," in *Progress in Speech Synthesis*. 1997, pp. 77–89.

[66] V Pagel, K Lenzo, and A. Black, "Letter-to-sound rules for accented lexicon compression," in *In Proc ICSLP*, 1998, 2015—2018.

[67] J. Suontausta and J. Häkkinen, "Decision tree based text-to-phoneme mapping for speech recognition," in *Proc. ICSLP*, 2000, pp. 831–834.

[68] L. Jiang, H.-W. Hon, and X. Huang, "Improvements on a trainable letter-to-sound converter," in *Proc Eurospeech*, 1997, pp. 605–608.

[69] S. F. Chen, "Conditional and joint models for grapheme-to-phoneme conversion," in *Proc Eurospeech*, 2003, pp. 2033–2036.

[70] H. Meng, S. Seneff, and V. Zue, "Phonological parsing for bi-directional letter-to-sound/sound-to-letter generation," in *Proc. HLT*, 1994, pp. 289–294.

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

[72] M. Dedina and H. Nusbaum, "PRONOUNCE: A program for pronunciation by analogy," *Computer Speech and Language*, vol. 5, no. 1, pp. 55–64, 1991.

[73] R. I. Damper and J. F. G. Eastmond, "Pronunciation by analogy: Impact of implementational choices on performance," *Language and Speech*, vol. 40, no. 1, pp. 1–23, 1997.

[74] R. I. Damper, C. Z. Stanbridge, and Y. Marchand, "A pronunciation-by-analogy module for the Festival text-to-speech synthesiser," in *Proc. SSW*, 2001, pp. 97–102.

[75] J. Bellegarda, "Unsupervised, language-independent grapheme-to-phoneme conversion by latent analogy," in *Proc. ICASSP*, 2003, pp. 244–247.

[76] M. Bisani and H. Ney, "Investigations on joint-multigram models for grapheme-to-phoneme conversion," in *Proc. ICSLP*, 2002, pp. 105–108.

[77] P. Vozila *et al.*, "Grapheme to phoneme conversion and dictionary verification using graphonemes," in *Proc Eurospeech*, 2003, pp. 2469–2472.

[78] M. Bisani and H. Ney, *Sequitur Github repository*, 2008. [Online]. Available: `https://github.com/sequitur-g2p/sequitur-g2p`.

[79] J. R. Novak *et al.*, "Improving WFST-based G2P conversion with alignment constraints and RNNLM n-best rescoring," in *Proc Interspeech*, 2012, pp. 2526–2529.

[80] J. R. Novak, N. Minematsu, and K. Hirose, "Phonetisaurus: Exploring grapheme-to-phoneme conversion with joint n-gram models in the WFST framework," *Natural Language Engineering*, vol. 22, no. 6, 907–938, 2016.

[81] S. Toshniwal and K. Livescu, "Jointly learning to align and convert graphemes to phonemes with neural attention models," in *Proc. SLT*, 2016, pp. 76–82.

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

[83] S. Yolchuyeva, G. Németh, and B. Gyires-Tóth, "Grapheme-to-phoneme conversion with convolutional neural networks," *Applied Sciences*, vol. 9, no. 6, 2019.

[84] M.-J. Chae *et al.*, "Convolutional sequence to sequence model with non-sequential greedy decoding for grapheme to phoneme conversion," in *Proc. ICASSP*, 2018, pp. 2486–2490.

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

[86] A. Bruguier, A. Bakhtin, and D. Sharma, "Dictionary augmented sequence-to-sequence neural network for grapheme to phoneme prediction," in *Proc. Interspeech*, 2018, pp. 3733–3737.

[87] D. Van Esch, M. Chua, and K. Rao, "Predicting pronunciations with syllabification and stress with recurrent neural networks," in *Proc. Interspeech*, 2016, pp. 2841–2845.

[88] H. Bleyan *et al.*, "Developing pronunciation models in new languages faster by exploiting common grapheme-to-phoneme correspondences across languages," in *Proc. Interspeech*, 2019, pp. 2100–2104.

[89] M. Yu *et al.*, "Multilingual grapheme-to-phoneme conversion with byte representation," in *Proc. ICASSP*, 2020, pp. 8234–8238.

[90] K. Gorman *et al.*, "The SIGMORPHON 2020 shared task on multilingual grapheme-to-phoneme conversion," in *Proc. SIGMORPHON*, 2020, pp. 40–50.

[91] B. Maison, S. Chen, and P. Cohen, "Pronunciation modeling for names of foreign origin," in *Proc. ASRU*, 2003, pp. 429–434.

[92] N. Cremelie and L. T. Bosch, "Improving the recognition of foreign names and non-native speech by combining multiple grapheme-to-phoneme converters," in *Proc. ITRW*, 2001, pp. 151–154.

[93] B. Réveil, J.-P. Martens, and H. van den Heuvel, "Improving proper name recognition by adding automatically learned pronunciation variants to the lexicon," in *Proc. LREC*, 2010, pp. 2149–2154.

[94] K. Rao, F. Peng, and F. Beaufays, "Automatic pronunciation verification for speech recognition," in *Proc. ICASSP*, 2015, pp. 5162–5166.

[95] S. Waxmonsky and S. Reddy, "G2P conversion of proper names using word origin information," in *Proc. NAACL*, 2012, pp. 367–371.

[96] A. Llitjos and A. W. Black, "Knowledge of language origin improves pronunciation accuracy of proper names," in *Proc. Eurospeech*, 2001, pp. 1919–1922.

[97] A. Font Llitjós and A. W. Black, "Evaluation and collection of proper name pronunciations online," in *Proc. LREC*, 2002, pp. 247–254.

[98] A. T. Rutherford, F. Peng, and F. Beaufays, "Pronunciation Learning for Named-Entities through Crowd-Sourcing," in *Proc. Interspeech*, 2014, pp. 1448–1452.

[99] Z. Kou *et al.*, "Fix it where it fails: Pronunciation learning by mining error corrections from speech logs," in *Proc. ICASSP*, 2015, pp. 4619–4623.

[100] T. Bruguier, F. Peng, and F. Beaufays, "Learning personalized pronunciations for contact names recognition," in *Proc. Interspeech*, 2016, pp. 3096–3100.

[101] A. Bruguier *et al.*, "Pronunciation learning with RNN-transducers," in *Proc. Interspeech*, 2017, pp. 2556–2560.

[102] T. Polyákova and A. Bonafonte, "Introducing nativization to Spanish TTS systems," *Speech Communication*, vol. 53, no. 8, pp. 1026–1041, 2011.

[103] J. Mendelson *et al.*, "Nativization of foreign names in TTS for automatic reading of world news in Swahili," in *Proc. Interspeech*, 2017, pp. 2188–2192.

[104] S. King, "Measuring a decade of progress in text-to-speech," *Loquens*, vol. 1, 2014.

[105] T. Merritt *et al.*, "Deep neural network-guided unit selection synthesis," in *Proc. ICASSP*, 2016, pp. 5145–5149.

[106] T. Capes *et al.*, "Siri on-device deep learning-guided unit selection text-to-speech system," in *Proc. Interspeech*, 2017, pp. 4011–4015.

[107] Y. Fan *et al.*, "TTS synthesis with bidirectional LSTM based recurrent neural networks," in *Proc. Interspeech*, 2014, pp. 1964–1968.

[108] H. Zen and H. Sak, "Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis," in *Proc. ICASSP*, 2015, pp. 4470–4474.

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

[110] X. Tan *et al.*, *A survey on neural speech synthesis*, 2021. arXiv: `2106.15561 [eess.AS]`.

[111] W. Ping *et al.*, "Deep Voice 3: Scaling text-to-speech with convolutional sequence learning," in *Proc. ICLR*, 2018.

[112] 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.

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

[114] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," in *Proc. ICASSP*, 2019, pp. 3617–3621.

[115] S. Kim *et al.*, "High fidelity speech synthesis with adversarial networks," in *Proc ICML*, 2019.

[116] K. Kumar *et al.*, "MelGAN: Generative adversarial networks for conditional waveform synthesis," in *Proc NeurIPS*, 2019.

[117] R. Yamamoto, E. Song, and J.-M. Kim, "Parallel WaveGAN: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," in *Proc. ICASSP*, 2020, pp. 6199–6203.

[118] J. Kong, J. Kim, and J. Bae, "HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis," in *Advances in Neural Information Processing Systems*, H. Larochelle *et al.*, Eds., vol. 33, Curran Associates, Inc., 2020, pp. 17 022–17 033.

[119] Z. Kong *et al.*, "DiffWave: A versatile diffusion model for audio synthesis," in *Proc. ICLR*, 2021.

[120] N. Chen *et al.*, "WaveGrad: Estimating gradients for waveform generation," in *Proc. ICLR*, 2021.

[121] Z. Mu, X. Yang, and Y. Dong, *Review of end-to-end speech synthesis technology based on deep learning*, 2021. arXiv: 2104.09995 [cs.SD].

[122] C. Miao *et al.*, "EfficientTTS: An efficient and high-quality text-to-speech architecture," in *Proc. ICML*, vol. 139, 2021, pp. 7700–7709.

[123] J. Kim, J. Kong, and J. Son, *Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech*, 2021. arXiv: 2106.06103 [cs.SD].

[124] W. Ping, K. Peng, and J. Chen, "ClariNet: Parallel wave generation in end-to-end text-to-speech," in *Proc. ICLR*, 2019.

[125] A. van den Oord *et al.*, *Wavenet: A generative model for raw audio*, 2016. arXiv: 1609.03499 [cs.SD].

[126] ——, "Parallel WaveNet: Fast high-fidelity speech synthesis," in *Proc. ICML*, 2018, pp. 3918–3926.

[127] M. Bińkowski *et al.*, "High fidelity speech synthesis with adversarial networks," in *Proc. ICLR*, 2020.

[128] H. Kawahara, "STRAIGHT, exploitation of the other aspect of vocoder: Perceptually isomorphic decomposition of speech sounds," *Acoustical Science and Technology*, vol. 27, no. 6, pp. 349–353, 2006.

[129] M. Morise, F. Yokomori, and K. Ozawa, "WORLD: A vocoder-based high-quality speech synthesis system for real-time applications," *IEICE Transactions on Information and Systems*, vol. E99.D, no. 7, pp. 1877–1884, 2016.

[130] J.-M. Valin and J. Skoglund, "LPCNet: Improving neural speech synthesis through linear prediction," in *Proc. ICASSP*, 2019, pp. 5891–5895.

[131] N. Li *et al.*, "Neural speech synthesis with Transformer network," in *Proc. AIII*, 2019.

[132] Y. Ren *et al.*, "FastSpeech: Fast, robust and controllable text to speech," in *Proc. NeurIPS*, 2019.

[133] X. Zhu *et al.*, "Pre-alignment guided attention for improving training efficiency and model stability in end-to-end speech synthesis," *IEEE Access*, vol. 7, pp. 65 955–65 964, 2019.

[134] 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.

[135] J. Shen *et al.*, "Non-attentive Tacotron: Robust and controllable neural TTS synthesis including unsupervised duration modeling," in *submission to ICLR*, 2021.

[136] T. Kenter, M. Sharma, and R. Clark, "Improving the prosody of RNN-based english text-to-speech synthesis by incorporating a BERT model," in *Interspeech*, 2020, pp. 4412–4416.

[137] S. Tyagi *et al.*, "Dynamic prosody generation for speech synthesis using linguistics-driven acoustic embedding selection," in *Proc. Interspeech*, 2020, pp. 4407–4411.

[138] A. Gibiansky *et al.*, "Deep Voice 2: Multi-speaker neural text-to-speech.," in *Proc. NIPS*, 2017, pp. 2962–2970.

[139] Y.-A. Chung *et al.*, "Semi-supervised training for improving data efficiency in end-to-end speech synthesis," in *Proc. ICASSP*, 2019, pp. 6940–6944.

[140] Y. Zhang *et al.*, "Learning to speak fluently in a foreign language: Multilingual speech synthesis and cross-language voice cloning," in *Proc. Interspeech*, 2019, pp. 2080–2084.

[141] S. Maiti, E. Marchi, and A. Conkie, "Generating multilingual voices using speaker space translation based on bilingual speaker data," in *Proc. ICASSP*, 2020, pp. 7624–7628.

[142] Y.-J. Chen *et al.*, "End-to-end text-to-speech for low-resource languages by cross-lingual transfer learning," in *Proc. Interspeech*, 2019, pp. 2075–2079.

[143] T. Yanagita, S. Sakti, and S. Nakamura, "Neural iTTS: Toward synthesizing speech in real-time with end-to-end neural text-to-speech framework," in *Proc. SSW*, 2019, pp. 183–188.

[144] B. Stephenson *et al.*, "What the future brings: Investigating the impact of lookahead for incremental neural TTS," in *Proc. Interspeech*, 2020, pp. 215–219.

[145] M. Ma *et al.*, "Incremental text-to-speech synthesis with prefix-to-prefix framework," in *Proc. EMNLP*, 2020, pp. 3886–3896.

[146] N. Ellinas *et al.*, "High quality streaming speech synthesis with low, sentence-length-independent latency," in *Proc. Interspeech*, 2020, pp. 2022–2026.

[147] O.-S. Arik *et al.*, "Deep Voice: Real-time neural text-to-speech," in *Proc. ICML*, 2017, pp. 195–204.

[148] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008.

[149] Y. Jia *et al.*, "PnG BERT: Augmented BERT on phonemes and graphemes for neural TTS," in *To Appear at Interspeech*, 2021.

[150] *eSpeak text to speech*, 2007. [Online]. Available: `http://espeak.sourceforge.net/`.

[151] *g2pE: A simple python module for english grapheme to phoneme conversion*, 2018. [Online]. Available: `https://github.com/Kyubyong/g2p`.

[152] *Pronunciation dictionaries for multiple languages*, 2017. [Online]. Available: `https://github.com/Kyubyong/pron_dictionaries`.

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

[154] D. Stanton, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," Talk at Google Speech Summit, 2018.

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

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

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

[158] C. Veaux, J. Yamagishi, and K. MacDonald, *VCTK corpus: English multi-speaker corpus*, 2019. [Online]. Available: `https://datashare.is.ed.ac.uk/handle/10283/2651`.

[159] H. Zen *et al.*, "LibriTTS: A corpus derived from LibriSpeech for text-to-speech," in *Proc. Interspeech*, 2019, pp. 1526–1530.

[160] J. Kominek and A. W. Black, "Learning pronunciation dictionaries: Language complexity and word selection strategies," in *Proc. NAACL*, 2006, pp. 232–239.

[161] K. Park and T. Mulc, "CSS10: A collection of single speaker speech datasets for 10 languages," in *Proc. Interspeech*, 2019, pp. 1566–1570.

[162] J. Kominek and A. W. Black, "The CMU Arctic speech databases," in *Proc. SSW*, 2004, pp. 223–224.

[163] G. Klein *et al.*, "OpenNMT: Open-source toolkit for neural machine translation," pp. 67–72, 2017.

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

[165] B. Hixon, E. Schneider, and S. L. Epstein, "Phonemic similarity metrics to compare pronunciation methods," in *Proc. Interspeech*, 2011, pp. 825–828.

[166] J. Fong *et al.*, "Investigating the robustness of sequence-to-sequence text-to-speech models to imperfectly-transcribed training data," in *Proc. Interspeech*, 2019, pp. 1546–1550.

[167] G. Leech, P. Rayson, and A. Wilson, *Frequency lists for Word Frequencies in Written and Spoken English: based on the British National Corpus*. 2001. [Online]. Available: `https://ucrel.lancs.ac.uk/bncfreq/flists.html`.

[168] P. T. Alan W Black and R. Caley., "The Festival speech synthesis system," Tech. Rep., 2014. [Online]. Available: `http://www.festvox.org/docs/manual-2.4.0/festival_toc.html`.

[169] "Method for the subjective assessment of intermediate quality level of coding systems," Tech. Rep. ITU Recommendation ITU-R BS.1534-1, 2003. [Online]. Available: `https://www.itu.int/dms_pubrec/itu-r/rec/bs/R-REC-BS.1534-3-201510-I!!PDF-E.pdf`.

[170] S. Shirali-Shahreza and G. Penn, "MOS naturalness and the quest for human-like speech," in *Proc. SLT*, 2018, pp. 346–352.

[171] K. Gorman, G. Mazovetskiy, and V. Nikolaev, "Improving homograph disambiguation with supervised machine learning," in *Proc. LREC*, 2018, pp. 1349–1352.

[172] X. Zhang *et al.*, "Acoustic data-driven lexicon learning based on a greedy pronunciation selection framework," in *Proc. Interspeech*, 2017, pp. 2541–2545.

[173] Y. Yasuda, X. Wang, and J. Yamagishi, "Investigation of learning abilities on linguistic features in sequence-to-sequence text-to-speech synthesis," *Computer Speech & Language*, vol. 67, pp. 101–183, 2021.

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

[175] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," in *ICML Workshop on Deep Learning*, 2015.

[176] J. Chung *et al.*, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *NIPS Workshop on Deep Learning*, 2014.

[177] J. Chevelu *et al.*, "How to compare TTS systems: A new subjective evaluation methodology focused on differences," in *Proc. Interspeech*, 2015, pp. 3481–3485.

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

[179] Y. Wu *et al.*, *Google's neural machine translation system: Bridging the gap between human and machine translation*, 2016. arXiv: `1609.08144 [cs.CL]`.

[180] A. Matthews, G. Neubig, and C. Dyer, "Using morphological knowledge in open-vocabulary neural language models," in *Proc. ACL*, 2018, pp. 1435–1445.

[181] D. Gowda *et al.*, "Multi-task multi-resolution char-to-BPE cross-attention decoder for end-to-end speech recognition," in *Proc. Interspeech*, 2019, pp. 2783–2787.

[182] W. Zhou *et al.*, "Acoustic data-driven subword modeling for end-to-end speech recognition," in *submission to Interspeech*, 2021.

[183] D. Renshaw and K. B. Hall, "Long short-term memory language models with additive morphological features for automatic speech recognition," in *Proc. ICASSP*, 2015, pp. 5246–5250.

[184] G. Montavon, W. Samek, and K.-R. Müller, "Methods for interpreting and understanding deep neural networks," *Digital Signal Processing*, vol. 73, pp. 1–15, 2018.

[185] Z. C. Lipton, "The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery.," *Queue*, vol. 16, no. 3, 31–57, 2018.

[186] Y. Belinkov and J. Glass, "Analyzing hidden representations in end-to-end automatic speech recognition systems," in *Proc. NIPS*, vol. 30, 2017, 2438–2448.

[187] A. Baevski, S. Schneider, and M. Auli, "vq-wav2vec: Self-supervised learning of discrete speech representations," in *Proc. ICLR*, 2020.

[188] G. C. Tal Linzen and A. Alishahi, "Proceedings of the 2018 EMNLP workshop BlackboxNLP: Analyzing and interpreting neural networks for NLP," in *Proc. BlackboxNLP*, 2018.

[189] A. Alishahi *et al.*, "Proceedings of the third BlackboxNLP workshop on analyzing and interpreting neural networks for NLP," in *Proc. BlackboxNLP*, 2020.

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

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

[192] L. Wang *et al.*, "Morphological segmentation with window LSTM neural networks," in *Proc. AAAI*, 2016, pp. 2842–2848.

[193] M. Bikmetova, "Grapheme-to-metaphoneme conversion for Unisyn and Combilex baseform transcription," M.S. thesis, University of Edinburgh, 2018.

[194] C. Northey, "Neural approaches to morphological decomposition for pronunciation learning," M.S. thesis, University of Edinburgh, 2019.

[195] P. Smit *et al.*, "Morfessor 2.0: Toolkit for statistical morphological segmentation," in *Proc. EACL*, 2014, pp. 21–24.

[196] T. Kudo and J. Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," in *Proc. EMNLP*, 2018, pp. 66–71.

[197] D. Jordan, "Linguistically-augmented approaches to G2P conversion," M.S. thesis, University of Edinburgh, 2019.

[198] R. Cotterell *et al.*, "The SIGMORPHON 2016 shared task: Morphological reinflection," in *Proc. SIGMORPHON*, 2016, pp. 10–22.

[199] R. Cotterell *et al.*, "The CoNLL–SIGMORPHON 2017 shared task: Universal morphological reinflection in 52 languages," in *Proc. SIGMORPHON*, 2017, pp. 1–30.

[200] R. Cotterell *et al.*, "The CoNLL–SIGMORPHON 2018 shared task: Universal morphological reinflection," in *Proc. SIGMORPHON*, 2018, pp. 1–27.

[201] A. D. McCarthy *et al.*, "The SIGMORPHON 2019 shared task: Morphological analysis in context and cross-lingual transfer for inflection," in *Proc. SIGMORPHON*, 2019, pp. 229–244.

[202] Fatchord, *Tacotron implementation*, Available: `https://github.com/fatchord/WaveRNN`, 2020.

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

[204] Z. Hodari, O. Watts, and S. King, "Using generative modelling to produce varied intonation for speech synthesis," in *Proc. SSW*, 2019, pp. 239–244.

[205] Z. Cai *et al.*, "Polyphone disambiguation for Mandarin Chinese using conditional neural network with multi-level embedding features," in *Proc. Interspeech*, 2019, pp. 2110–2114.

[206] D. Dai *et al.*, "Disambiguation of Chinese polyphones in an end-to-end framework with semantic features extracted by pre-trained BERT."

[207] B. Yang, J. Zhong, and S. Liu, "Pre-trained text representations for improving front-end text processing in Mandarin text-to-speech synthesis," in *Proc. Interspeech*, 2019, pp. 4480–4484.

[208] A. Viehoff, "Linguistic augmentation in Kiswahili text-to-speech synthesis," M.S. thesis, University of Edinburgh, 2020.

[209] A. Bruguier, H. Zen, and A. Arkhangorodsky, "Sequence-to-sequence neural network model with 2D attention for learning Japanese pitch accents," in *Proc. Interspeech*, 2018, pp. 1284–1287.

[210] Y. Yasuda *et al.*, "Investigation of enhanced tacotron text-to-speech synthesis systems with self-attention for pitch accent language," in *Proc. ICASSP*, 2019, pp. 6905–6909.

[211] J. Pontes and S. Furui, "Predicting the phonetic realizations of word-final consonants in context – A challenge for French grapheme-to-phoneme converters," *Speech Communication*, vol. 52, no. 10, pp. 847–862, 2010.

[212] A. Greefhorst and A. Bosch, "Predicting liaison: An example-based approach," *Traitement Automatique des Langues*, vol. 57, pp. 13–32, Jan. 2016.

[213] J. Durand and C. Lyche, "French liaison in the light of corpus data," *Journal of French Language Studies*, vol. 18, no. 1, pp. 33–66, 2008.

[214] B. New *et al.*, "Lexique 2: A new French lexical database," *Behavior Research Methods, Instruments, & Computers*, vol. 36, no. 3, pp. 516–524, 2004.

[215] F. Béchet, "LIA PHON: Un systeme complet de phonétisation de textes," *Traitement automatique des langues*, vol. 42, no. 1, pp. 47–67, 2001.

[216] J. M. Kalmbach, *Guide de prononciation française pour apprenants finnophones*. University of Jyväskylä, 2018, p. 20. [Online]. Available: `http://research.jyu.fi/phonfr/20.html`.

[217] K. Toutanova *et al.*, "Feature-rich part-of-speech tagging with a cyclic dependency network," in *Proc. NAACL*, 2003, pp. 173–180.

[218] I. Steiner and S. L. Maguer, "Creating new language and voice components for the updated MaryTTS text-to-speech synthesis platform," in *Proc. LREC*, 2018, pp. 3171–3175.

[219] B. Naderi and R. Cutler, "An open source implementation of ITU-T Recommendation P.808 with validation," in *Interspeech*, 2020, pp. 2862–2866.

[220] F. Jimenez, B. Naderi, and S. Moller, "Effect of environmental noise in speech quality assessment studies using crowdsourcing," in *Proc. QoMEX*, 2020.

[221] R. Jiménez, L. Gallardo, and S. Möller, "Outliers detection vs. control questions to ensure reliable results in crowdsourcing.: A speech quality assessment case study," in *Proc. WWW*, 2018, 1127–1130.

[222] R. Jiménez *et al.*, "Intra- and inter-rater agreement in a subjective speech quality assessment task in crowdsourcing," in *Proc. WWW*, 2019, 1138–1143.

[223] A. Rix *et al.*, "Perceptual evaluation of speech quality (PESQ): A new method for speech quality assessment of telephone networks and codecs," vol. 2, 2001, pp. 749–752.

[224] R. Kubichek, "Mel-cepstral distance measure for objective speech quality assessment," in *Proc. PACRIM*, 1993, pp. 125–128.

[225] M. Chinen *et al.*, "ViSQOL v3: An open source production ready objective speech and audio metric," in *Proc. QoMEX*, 2020.

[226] R. Gupta, A. Avila, and T. Falk, "Towards a neuro-inspired no-reference instrumental quality measure for text-to-speech systems," in *Proc. QoMEX*, 2018.

[227] A. Avila *et al.*, "Non-intrusive speech quality assessment using neural networks," in *Proc. ICASSP*, 2019, pp. 631–635.

[228] J. Williams *et al.*, "Comparison of speech representations for automatic quality estimation in multi-speaker text-to-speech synthesis," in *Proc. Speaker Odyssey*, 2020, pp. 222–229.

[229] C. Lo *et al.*, "MOSNet: Deep learning-based objective assessment for voice conversion," in *Proc. of Interspeech*, 2019, pp. 1541–1545.

[230] Y. Choi, Y. Jung, and H. Kim, "Deep MOS predictor for synthetic speech using cluster-based modeling," in *Proc. of Interspeech*, 2020.

[231] C. Valentini-Botinhao, J. Yamagishi, and S. King, "Can objective measures predict the intelligibility of modified HMM-based synthetic speech in noise?" In *Proc. Interspeech*, 2011, pp. 1837–1840.

[232] M. Cooke, C. Mayo, and C. Valentini-Botinhao, "Intelligibility-enhancing speech modifications: The Hurricane Challenge," English, in *Proc. Interspeech*, 2013, pp. 3552–3556.

[233] K. Arai *et al.*, "Predicting speech intelligibility of enhanced speech using phone accuracy of DNN-based ASR system," in *Proc. Interspeech*, 2019, pp. 4275–4279.

[234] M. Ribeiro *et al.*, "TaL: A synchronised multi-speaker corpus of ultrasound tongue imaging, audio, and lip videos," in *Proc. SLT*, 2021, pp. 1109–1116.

[235] R. Vích, J. Nouza, and M. Vondra, "Automatic speech recognition used for intelligibility assessment of text-to-speech systems," in *Verbal and Nonverbal Features of Human-Human and Human-Machine Interaction*. 2008, pp. 136–148.

[236] T. Godambe *et al.*, "Developing a unit selection voice given audio without corresponding text," *EURASIP*, vol. 2016, no. 1, Dec. 2016.

[237] K. Lee, E. Cooper, and J. Hirschberg, "A comparison of speaker-based and utterance-based data selection for text-to-speech synthesis," in *Proc. Interspeech*, 2018, pp. 2873–2877.

[238] F. Hinterleitner *et al.*, "Comparison of approaches for instrumentally predicting the quality of text-to-speech systems: Data from Blizzard Challenges 2008 and 2009," pp. 1325–1328, 2010.

[239] C. Norrenbrock *et al.*, "Towards perceptual quality modeling of synthesized audiobooks – Blizzard Challenge 2012," in *Proc. of Blizzard Challenge Workshop*, 2012.

[240] R. Ullmann *et al.*, "Objective intelligibility assessment of text-to-speech systems through utterance verification," in *Proc. of Interspeech*, 2015, pp. 3501–3505.

[241] L. Latacz and W. Verhelst, "Double-ended prediction of the naturalness ratings of the Blizzard Challenge 2008-2013," in *Proc. of Interspeech*, 2015, pp. 3486–3490.

[242] T. Yoshimura *et al.*, "A hierarchical predictor of synthetic speech naturalness using neural networks," in *Proc of Interspeech*, 2016, pp. 342–346.

[243] S. King and V. Karaiskos, "The Blizzard Challenge 2011," in *Proc. Blizzard Challenge Workshop*, 2011. [Online]. Available: `http://www.festvox.org/blizzard/bc2011/summary_Blizzard2011.pdf`.

[244] ——, "The Blizzard Challenge 2012," in *Proc. Blizzard Challenge Workshop*, 2012. [Online]. Available: `http://www.festvox.org/blizzard/bc2012/summary_Blizzard2012.pdf`.

[245] ——, "The Blizzard Challenge 2013," in *Proc. of Blizzard Challenge Workshop*, 2013. [Online]. Available: `http://www.festvox.org/blizzard/bc2013/summary_Blizzard2013.pdf`.

[246] ——, "The Blizzard Challenge 2016," in *Proc. of Blizzard Challenge Workshop*, 2016. [Online]. Available: `http://www.festvox.org/blizzard/bc2016/blizzard2016_overview_paper.pdf`.

[247] S. King, L. Wihlborg, and W. Guo, "The Blizzard Challenge 2017," in *Proc. of Blizzard Challenge Workshop*, 2017. [Online]. Available: `http://www.festvox.org/blizzard/bc2018/blizzard2018_overview_paper.pdf`.

[248] S. King *et al.*, "The Blizzard Challenge 2018," in *Proc. of Blizzard Challenge Workshop*, 2018. [Online]. Available: `http://www.festvox.org/blizzard/bc2018/blizzard2018_overview_paper.pdf`.

[249] V. Panayotov *et al.*, "Librispeech: An ASR corpus based on public domain audio books," *Proc. ICASSP*, pp. 5206–5210, 2015.

[250] A. Baby *et al.*, *An ASR guided speech intelligibility measure for TTS model selection*, 2020. arXiv: `2006.01463 [cs.SD]`.

[251] B. Efron and R. Tibshirani, "Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy," *Statistical Science*, vol. 1, no. 1, pp. 54–75, 1986.

[252] M. Bisani and H. Ney, "Bootstrap estimates for confidence intervals in ASR performance evaluation," 2004, pp. 409–411.

[253] M. Kendall, "The treatment of ties in ranking problems," *Biometrika*, vol. 33, no. 3, pp. 239–251, Nov. 1945.

[254] Z. Yi *et al.*, "Voice Conversion Challenge 2020: Intra-lingual semi-parallel and cross-lingual voice conversion," in *Proc. Voice Conversion Challenge*, 2020, pp. 80–98.

[255] E. Pusateri *et al.*, "A mostly data-driven approach to inverse text normalization," in *Proc. Interspeech*, 2017, pp. 2784–2788.

[256] Y. Zhang *et al.*, *Nemo inverse text normalization: From development to production*, 2021. arXiv: `2104.05055 [cs.CL]`.

[257] K. Hu *et al.*, "Phoneme-based contextualization for cross-lingual speech recognition in end-to-end models," in *Proc. Interspeech*, 2019, pp. 2155–2159.

[258] M. N. Sundararaman, A. Kumar, and J. Vepa, *Phoneme-BERT: Joint language modelling of phoneme sequence and ASR transcript*, 2021. arXiv: `2102.00804 [eess.AS]`.

[259] D. Le *et al.*, "G2G: TTS-driven pronunciation learning for graphemic hybrid ASR," in *Proc. ICASSP*, 2020, pp. 6869–6873.

[260] T. Okamoto *et al.*, "Tacotron-based acoustic model using phoneme alignment for practical neural text-to-speech systems," in *Proc. ASRU*, 2019, pp. 214–221.

[261] S. Panchapagesan *et al.*, “Multi-task learning and weighted cross-entropy for dnn-based keyword spotting,” in *Proc. Interspeech*, 2016, pp. 760–764.

[262] J. Hou *et al.*, “Mining effective negative training samples for keyword spotting,” in *Proc. ICASSP*, 2020, pp. 7444–7448.

[263] C. Shan *et al.*, “Attention-based end-to-end models for small-footprint keyword spotting,” in *Proc. Interspeech*, 2018, pp. 2037–2041.

[264] S. Myer and V. S. Tomar, “Efficient keyword spotting using time delay neural networks,” in *Proc. Interspeech*, 2018, pp. 1264–1268.

[265] J. Fong, J. Taylor, and S. King, “Testing the limits of representation mixing for pronunciation correction in end-to-end speech synthesis,” in *Proc. Interspeech*, 2020, pp. 4019–4023.

[266] A. Bruguier *et al.*, “Phoebe: Pronunciation-aware contextualization for end-to-end speech recognition,” in *Proc. ICASSP*, 2019, pp. 6171–6175.

[267] G. Gershwin and I. Gershwin, *Lyrics to "let’s call the whole thing off"*, 1937. [Online]. Available: `https://genius.com/Fred-astaire-lets-call-the-whole-thing-off-lyrics#about`.