Chapter

Generating the Voice of the Interactive Virtual Assistant

Adriana Stan and Beáta Lőrincz

Abstract

This chapter introduces an overview of the current approaches for generating spoken content using text-to-speech synthesis (TTS) systems, and thus the voice of an Interactive Virtual Assistant (IVA). The overview builds upon the issues which make spoken content generation a non-trivial task, and introduces the two main components of a TTS system: text processing and acoustic modelling. It then focuses on providing the reader with the minimally required scientific details of the terminology and methods involved in speech synthesis, yet with sufficient knowledge so as to be able to make the initial decisions regarding the choice of technology for the vocal identity of the IVA. The speech synthesis methodologies' description begins with the basic, easy to run, low-requirement rule-based synthesis, and ends up within the state-of-the-art deep learning landscape. To bring this extremely complex and extensive research field closer to commercial deployment, an extensive indexing of the readily and freely available resources and tools required to build a TTS system is provided. Quality evaluation methods and open research problems are, as well, highlighted at end of the chapter.

Keywords: text-to-speech synthesis, text processing, deep learning, interactive virtual assistant

1. Introduction

Generating the voice of an interactive virtual assistant (IVA) is performed by the so called *text-to-speech synthesis (TTS)* systems. A TTS system takes raw text as input and converts it into an acoustic signal or waveform, through a series of intermediate steps. The synthesised speech commonly pertains to a single, pre-defined speaker, and should be as natural and as intelligible as human speech. An overview of the main components of a TTS system is shown in **Figure 1**.

At first sight this seems like a straightforward mapping of each character in the input text to its acoustic realisation. However, there are numerous technical issues

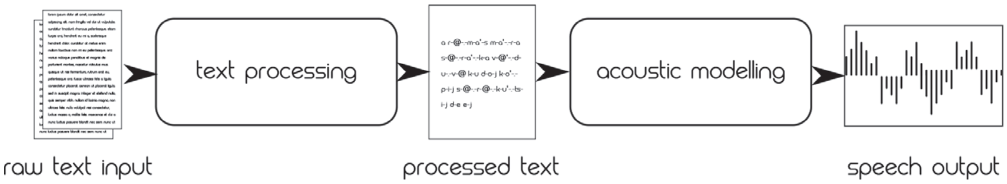

Figure 1.
Overview of a text-to-speech synthesis system's main components.

which make natural speech synthesis an extremely complex problem, with some of the most important ones being indexed below:

the written language is a discrete, compressed representation of the spoken language aimed at transferring a message, irrespective of other factors pertaining to the speaker’s identity, emotional state, etc. Also, in almost any language, the written symbols are not truly informative of their pronunciation, with the most notable example being English. The pronunciation of a letter or sequence of letters which yield a single sound is called a *phone*. One exception here is the Korean alphabet for which the symbols approximate the position of the articulator organs, and was introduced in 1443 by King Sejong the Great to increase the literacy among the Korean population. But for most languages, the so called orthographic transparency is rather opaque;

the human ear is highly adapted to the frequency regions in which the relevant information from speech resides (i.e. 50–8000 Hz). Any slight changes to what is considered to be natural speech, any artefacts, or unnatural sequences present in a waveform deemed to contain spoken content, will be immediately detected by the listener;

speaker and speech variability is a result of the uniqueness of each individual. This means that there are no two persons having the same voice timbre or pronouncing the same word in a similar manner. Even more so, one person will never utter a word or a fixed message in an exactly identical manner even when the repetitions are consecutive;

co-articulation effects derive from the articulator organs’ inertial movement. There are no abrupt transitions between sounds and, with very few exceptions, it is very hard to determine the exact boundary of each sound. Another result of the co-articulation is the presence of reductions or modifications in the spoken form of a word or sequence of words, derived from the impossibility or hardship of uttering a smooth transition between some particular phone pairs;

prosody is defined as the rhythm and melody or intonation of an utterance. The prosody is again related to the speaker’s individuality, cultural heritage, education and emotional state. There are no clear systems which describe the prosody of a spoken message, and one’s person understanding of, for example, portraying an angry state of mind is completely different from another;

no fixed set of measurable factors define a speaker’s identity and speaking characteristics. Therefore, when wanting to reproduce one’s voice the only way to do this for now is to record that person and extract statistical information from the acoustic signal;

no objective measure correlates the physical representation of a speech signal with the perceptual evaluation of a synthesised speech’s quality and/or appropriateness.

The problems listed above have been solved, to some extent, in TTS systems by employing high-level machine learning algorithms, developing large expert resources or by limiting the applicability and use-case scenarios for the synthesised speech. In the following sections we describe each of the main components of a TTS system, with an emphasis on the acoustic modelling part which poses the greatest problems as of yet. We also index some of the freely available resources and tools which can aid a fast development of a synthesis system for commercial IVAs in a dedicated section of the chapter, and conclude with the discussion of some open problems in the final section.

2. Speech processing fundamentals

Before diving into the text-to-speech synthesis components, it is important to define a basic set of terms related to digital speech processing. A complete overview of this domain is beyond the scope of this chapter, and we shall only refer to the terms used to describe the systems in the following sections.

Speech is the result of the air exhaled from the lungs modulated by the articulator organs and their instantaneous or transitioning position: vocal cords, larynx, pharynx, oral cavity, palate, tongue, teeth, jaw, lips and nasal cavity. By modulation we refer to the changes suffered by the air stream as it encounters these organs. One of the most important organs in speech are the vocal cords, as they determine the periodicity of the speech signal by quickly opening and closing as the air passes through. The vocal cords are used in the generation of vowels and voiced consonant sounds [1]. The perceived result of this periodicity is called the *pitch*, and its objective measure is called *fundamental frequency*, commonly abbreviated F_0 [2]. The slight difference between pitch and F_0 is better explained by the auditory illusion of the *missing fundamental* [3] where the measured fundamental frequency differs from the perceived pitch. Commonly, the terms are used interchangeably, but readers should be aware of this small difference. The pitch variation over time in the speech signal gives the melody or intonation of the spoken content. Another important definition is that of *vocal tract* which refers to all articulators positioned above the vocal cords. The resonance frequencies of the vocal tract are called *formant frequencies*. Three formants are commonly measured and noted as F_1, F_2 and F_3.

Looking into the time domain, as a result of the articulator movement, the speech signal is not stationary, and its characteristics evolve through time. The smallest time interval in which the speech signal is considered to be *quasi-stationary* is 20–40 msec. This interval determines the so-called *frame-level analysis* or *windowing* of the speech signal, in which the signal is segmented and analysed at more granular time scales for the resulting analysis to adhere to the digital signal processing theorems and fundamentals [4].

The *spectrum* or *instantaneous spectrum* is the result of decomposing the speech signal into its frequency components through Fourier analysis [5] on a frame-by-frame basis. Visualising the evolution of the spectrum through time yields the *spectrogram*. Because the human ear has a non-linear frequency response, the linear spectrum is commonly transformed into the *Mel spectrum*, where the Mel frequencies are a non-linear transformation of the frequency domain pertaining to the pitches judged by listeners to be equal in distance one from another. Frequency domain analysis is omnipresent in all speech related applications, and Mel spectrograms are the most common representations of the speech signal in the neural network-based synthesis.

One other frequency-derived representation of the speech is the *cepstral* [6] representation which is a transform of the spectrum aimed at separating the vocal tract and the vocal cord (or glottal) contributions from the speech signal. It is based on homomorphic and decorrelation operations.

3. Text processing

Text processing or *front-end processing* represents the mechanism of generating supplemental information from the raw input text. This information should yield a representation which is hypothetically closer and more relevant to the acoustic realisation of the text, and therefore tightens the gap between the two domains. Depending on the targeted language, this task is more or less complex [2]. A list of the common front-end processing steps is given below:

text tokenisation splits the input text into syntactically meaningful chunks i.e. phrases sentences and words. Languages which do not have a word separator such as Chinese or Japanese pose additional complexity for this task [7];

diacritic restoration - in languages with diacritic symbols it might be the case that the user does not type these symbols and this leads to an incorrect spoken sequence [8]. The diacritic restoration refers to adding the diacritic symbols back into the text so that the intended meaning is preserved;

text normalisation converts written expressions into their ‘“spoken” forms e.g. $3.16 is converted into “three dollars sixteen cents.” or 911 is converted into “nine one one” and not “nine hundred eleven” [9]. An additional problem is caused by languages which have genders assigned to nouns e.g. in Romanian “21 oi = douăzeci şi *una* de oi” (en. twenty one sheep–feminine) versus “21 cai = douăzeci şi *unu* de cai” (en. twenty one horses-masculine);

part-of-speech tagging (POS) assigns a part-of-speech (i.e. noun, verb, adverb, adjective, etc.) to each word in the input sequence. The POS is important to disambiguate non-homophone homographs. These are words which are spelled the same but pronounced differently based on their POS (e.g. *bow* - to bend down/the front of a boat/tied loops). POS are also essential for placing the accent or focus of an utterance on the correct word or word sequence [10];

lexical stress marking - the lexical stress pertains to the syllable within a word which is more prominent [11]. There are however languages for which this notion is quite elusive such as French or Spanish. Yet in English a stress-timed language assigning the correct stress to each word is essential for conveying the correct message. Along with the POS the lexical stress also helps disambiguate non-homophone homographs in the spoken content. There are also phoneticians who would mark a secondary and tertiary stress but for speech synthesis the primary stress should be enough as the secondary does not affect the meaning but rather the naturalness or emphasis of the speech;

syllabification - syllables represent the base unit of co-articulation and determine the rhythm of speech [12]. Again different languages pose different problems and languages such as Japanese rely on syllables for their alphabetic inventory. As a general rule every syllable has only one vowel sound but can be accompanied by semi-vowels. Compound words generally do not follow the general rules such that prefixes and suffixes will be pronounced as a single syllable;

phonetic transcription is the final result of all the steps above. Meaning that by knowing the POS the lexical stress and syllabification of a word the exact pronunciation can be derived [13]. The phones are a set of symbols corresponding to an individual articulatory target position in a language or otherwise put it is the fixed sound alphabet of a language. This alphabet determines how each sequence of letters should be pronounced. Yet this is not always the case and the concept of orthographic transparency determines the ease with which a reader can utter a written text in a particular language;

prosodic labels, phrase breaks - with all the lexical information in place there is still the issue of emphasising the correct words as per intent of the writer. The accent and pauses in speech are very important and can make the message decoding a very complex task or an easier one with the information being able to be faster assimilated by the listener. There is quite a lot of debate on how the prosody should be marked in text and if it should be [14]. There is definitely some markings in the form of punctuation signs yet there is a huge gap between the text and the spoken output. However public speaking coaching puts a large weight on the prosodic aspect of the speech and therefore captivating the listeners attention through non-verbal queues;

word/character embeddings - are the result of converting the words or characters in the text into a numeric representation which should encompass more information about their identity pronunciation syntax or meaning than the surface form does. Embeddings are learnt from large text corpora and are language dependent. Some of the algorithms used to build such representations are: Word2Vec [15] GloVe [16] ELMo [17] and BERT [18].

4. Acoustic modelling

The acoustic modelling or *back-end processing* part refers to the methods which convert the desired input text sequence into a speech waveform. Some of the earliest proofs of so-called talking heads are mentioned by Aurrilac (1003 A.D.), Albert Magnus (1198–1280) or Roger Bacon (1214–1294). The first electronic synthesiser was the VODER (Voice Operation DEmonstratoR) created by Homer Dudley at Bell Laboratories in 1939. The VODER was able to generate speech by tediously operating a keyboard and foot pedals to control a series of digital filters.

Coming to the more recent developments, and based on the main method of generating the speech signal, speech synthesis systems can be classified into **rule-based** and **corpus-based** methods. In rule-based methods, similar to the VODER, the sound is generated by a fixed, pre-computed set of parameters.

Corpus-based methods, on the other hand, use a set of speech recordings to generate the synthetic output or to derive statistical parameters from the analysis of the spoken content. It can be argued that using pre-recorded samples is not in itself synthesis, but rather a speech collage. In this sense Taylor gives a different definition of speech synthesis: *"the output of a spoken utterance from a resource in which it has not been prior spoken"* [2].

4.1 Rule-based synthesis

Formant synthesis is one of the first digital methods of speech generation. It is still used today, especially by phoneticians who study various spoken language phenomena. The method uses the approximation of several speech parameters (commonly the F_0 and formant frequencies) for each phone in a language, and also how these parameters vary when transitioning from one phone to the next one [19]. The most representative model of formant synthesis is the one described by [20], which later evolved into the commercial system of MITalk [21]. There are around 40 parameters which describe the formants and their respective bandwidths, and also a series of frequencies for nasals or glottal resonators.

The advantages of formant synthesis are related to the good intelligibility even at high speeds, and its very low computation and memory requirements, making it easy to deploy on limited resource devices. The major drawback of this type of synthesis is, of course, its low quality and robotic sound, and also the fact that for high-pitched outputs, the formant tracking mechanisms can fail to determine the correct values.

Articulatory synthesis uses mechanical and acoustic models of speech production [1]. The physiological effects such as the movement of the tongue, lips, jaw, and the dynamics of the vocal tract and glottis are modelled. For example, [22] uses lip opening, glottal area, opening of nasal cavities, constriction of tongue, and rate between expansion and contraction of the vocal tract along with the first four formant frequencies. Magnetic resonance imaging offers some more insight into the muscle movement [23], yet the complexity of this type of synthesis makes it rather unfeasible for high naturalness and commercial deployment. One exception in the project GNUSpeech [24] but its results are still poor compared to what corpus-based synthesis is able to achieve nowadays.

4.2 Corpus-based synthesis

4.2.1 Concatenative synthesis

As the name entails, concantenative synthesis is a method of producing spoken content by concatenating pre-recorded speech samples. In its most basic form, a concatenative synthesis system contains recordings of all the words needed to be uttered, which are then combined in a very limited vocabulary scenario. For example, in a rudimentary IVA, it will combine the typed-in phone number of a customer by combining pre-recorded digits. Of course, in a large vocabulary, open-domain system, pre-recording all the words in a language is unfeasible. The solution to this problem is to find a smaller set of acoustic units which can be then combined into any spoken phrase. Based on the type of segment stored in the recorded database, the concatenative synthesis is either **fixed inventory** – segments in the database have the same length, or **variable inventory or unit selection** – segments have variable length. As the basic acoustic unit of any language is its phone set, a first open-domain fixed inventory concatenative synthesis made use of *diphones* [25, 26]. A diphone is the acoustic unit spanning from the middle of a phone to the middle of the next one in adjoining phone pairs. Although this yields a much larger acoustic inventory, the diphones are a better choice than phones because they can model the co-articulation effects. For a primitive diphone concatenation system, the recorded speech corpus would include a single repetition of all the diphones in a language. More elaborate systems use diphones in different context (e.g. beginning, middle or end of a word) and with different prosodic events (e.g. accent, variable duration etc.). Another type of fixed inventory system is based on the use of *syllables* as the concatenation unit [27–29]. Some theories state that the basic unit of speech is the syllable and, therefore, the co-articulation effects between them is minimum [30], but the speech database is hard to design. The average number of unique syllables in one language is in the order of thousands.

A natural evolution of the fixed inventory synthesis is the variable length inventory, or unit selection [31, 32]. In unit selection, the recorded corpus includes segments as small as half-phones and go up to short common phrases. The speech database is either stored as-is, or as a set of parameters describing the exact acoustic waveform. The speech corpus, therefore, needs to be very accurately annotated with information regarding the exact phonetic content and boundaries, lexical stress, syllabification, lexical focus and prosodic trends or patterns (e.g. questions, exclamation, statements). The combination of the speech units into the output spoken phrase is done in an iterative manner, by selecting the best speech segments which minimise a global cost function [31] composed of: a *target cost* - measuring how well a sequence of units matches the desired output sequence, and a *concatenation cost* - measuring how well a sequence of units will be joined together and thus avoid the majority of the concatenation artefacts.

Although this type of synthesis is almost 30 years old, it is still present in many commercial applications. However, it poses some design problems, such as: the need for a very large manually segmented and annotated speech corpus; the control of prosody is hard to achieve if the corpus does not contain all the prosodic events needed to synthesise the desired output; changing the speaker identity requires the database recording and processing to be started from scratch; and there are quite a lot of concatenation artefacts present in the output speech making it unnatural, but which have, in some cases, been solved by using a hybrid approach [33].

4.2.2 Statistical-parametric synthesis

Because concatenative synthesis is not very flexible in terms of prosody and speaker identity, in 1989 a first model of statistical-parametric synthesis based on Hidden Markov Models (HMMs) was introduced [34]. The model is parametric because it does not use individual stored speech samples, but rather parameterises the waveform. And it is statistical because it describes the extracted parameters using statistics averaged across the same phonetic identity in the training data [35]. However this first approach did not attract the attention of the specialists because of its highly unnatural output. But in 2005, the HMM-based Speech Synthesis System (HTS) [36] solved part of the initial problems, and the method became the main approach in the research community with most of its studies aiming at fast speaker adaptation [37] and expressivity [38]. In HTS, a 3 state HMM models the statistics of the acoustic parameters of the phones present in the training set. The phones are clustered based on their identity, but also on other contextual factors, such as the previous and next phone identity, the number of syllables in the current word, the part-of-speech of the current word, the number of words in the sentence, or the number of sentences in a phrase, etc. This context clustering is commonly performed with the help of decision trees and ensures that the statistics are extracted from a sufficient number of exemplars. At synthesis time, the text is converted in a context aware complex label and drives the selection of the HMM states and their transitions. The modelled parameters are generally derived from the source-filter model of speech production [1]. One of the most common vocoders used in HTS is STRAIGHT [39] and it parameterises the speech waveform into F_0, Mel cepstral and aperiodicity coefficients. A less performant, yet open vocoder is WORLD [40]. A comparison of several vocoders used for statistical parametric speech synthesis is presented in [41].

There are several advantages for the statistical-parametric synthesis, such as: the small footprint necessary to store speech information; automatic clustering of speech information–removes the problems of hand-written rules; generalisation–even if for a certain phoneme context there is not enough training data, the phone will be clustered along with similar parameter characteristics; flexibility–the trained models can be easily adapted to other speakers or voice characteristics with minimum amount of adaptation data. However, the parameter averaging yields the so-called *buzziness* and low speaker similarity of the output speech, and for this reason the HTS system has not truly made its way into the commercial applications.

4.2.3 Neural synthesis

In 1943, McCulloch and Pitts [42] introduced the first computational model for artificial neural networks (ANN). And although the incipient ANNs have been successfully applied in multiple research areas, including TTS [43], their learning power comes from the ability to stack multiple neural layers between the input and output. However, it was not until 2006 that the hardware and algorithmic solutions enabled adding multiple layers and making the learning process stable. In 2006, Geoffrey Hinton and his team published a series of scientific papers [44, 45] showing how a many-layered neural network could be effectively pre-trained one layer at a time. These remarkable results set the trend for all automatic machine learning algorithms in the following years, and are the bases of the **deep neural network (DNN)** research field. Nowadays, there are very few machine learning applications which do not cite the DNNs as attaining the state-of-the-art results and performances.

In text-to-speech synthesis, the progression from HMMs to DNNs was gradual. Some of the first impacting studies are those of Ling et al. [46] and Zen et al. [47]. Both papers substitute parts of the HMM-based architecture, yet model the audio on a frame-by-frame basis, maintaining the statistical-parametric approach, and also use the same contextual factors in the text processing part. The first open source tool to implement the DNN-based statistical-parametric synthesis is Merlin [48]. A comparison of the improvements achieved by the DNNs compared to HMMs is presented in [49]. However, these methods still rely on a time-aligned set of text features and their acoustic realisations, which requires a very good frame-level aligner systems, usually an HMM-based one. Also, the sequential nature of speech is only marginally modelled through the contextual factors and not within the model itself, while the text still needs to be processed with expert linguistic automated tools which are rarely available in non-mainstream languages.

An intermediate system which replaces all the components in a TTS pipeline with neural networks is that of [50], but it does not incorporate a single end-to-end network. The first study which removes the above dependencies, and models the speech synthesis process as a sequence-to-sequence recurrent network-based architecture is that of Wang et al. [51]. The architecture was able to *"synthesise fairly intelligible speech"* and was the precursor of the more elaborate Char2Wav [52] and Tacotron [53] systems. Both Char2Wav and Tacotron model the TTS generation as a two step process: the first one takes the input text string and converts it into a spectrogram, and the second one, also called the *vocoder*, takes the spectrogram and converts it into a waveform, either in a deterministic manner [54], or with the help of a different neural network [55]. These two synthesis systems were also the first to alleviate the need for more elaborate text representations, and derived them as an inherent learning process, setting the first stepping stones towards true end-to-end speech synthesis [56]. However, for phonetically rich languages it is common to train the models on phonetically transcribed text, and also to augment the input text with additional linguistic information such as part-of-speech tags which can enhance the naturalness of the output speech [57, 58].

Starting with the publication of Tacotron, the DNN-based speech synthesis research and development area has seen an enormous interest from both the academia and the commercial sides. Most focus has been granted on generating extremely high quality speech, but also to the reduction of the computational requirements and generation speed–which in the DNN domain is called *inference* speed. A major breakthrough was obtained by the second version of Tacotron, Tacotron 2 [59], which achieved naturalness scores very close to human speech. However, both systems' architectures involve attention-based recurrent auto-regressive processes which make the inference step very slow and prone to instability issues, such as word skipping, deletions or repetitions. Also, the recurrent neural networks (RNNs) are known to have high demands in terms of data availability and training time. So that, the next step in DNN-based TTS was the introduction of CNNs, in systems such as DC-TTS [60], DeepVoice 3 [61], ClariNet [62], or ParaNet [63]. The CNNs enable a much better data and training efficiency and also a much faster inference speed through parallel processing. And also, recently, the research community started to look into ways of replacing the auto-regressive attention-based generation, and incorporated duration prediction models which stabilise the output and enable a much faster parallel inference of the output speech [64, 65].

Inspired by the success of the Transformer network [66] in text processing, TTS systems have adopted this architecture as well. Transformer based models include Transformer-TTS [67], FastSpeech [68], FastSpeech 2 [69], AlignTTS [70], JDI-T [71], MultiSpeech [72], or Reformer-TTS [73]. Transformer-based architectures improve the training time requirements, and are capable of modelling longer term dependencies present in the text and speech data.

As the naturalness of the output synthetic speech became very high-quality, researchers started to look into ways of easily controlling the different factors of the synthetic speech, such as duration or style. The go-to solution for this are the Variational AutoEncoders (VAEs) and their variations, which enable the disentanglement of the latent representations, and thus a better control of the inferred features [74–78]. There were also a few approaches including Generative Adversarial Networks (GANs), such as GAN-TTS [79] or [80], but due to the fact that GANs are known to pose great training problems, this direction was not that much explored in the context of TTS.

A common problem in all generative modelling irrespective of deep learning methodologies, is the fact that the true probability distribution of the training data is not directly learned or accessible. In 2015, Rezende et al. [81] introduced the normalising flows (NFs) concept. NFs estimate the true probability distribution of the data by deriving it from a simple distribution through a series of invertible transforms. The invertible transforms make it easy to project a measured data point into the latent space and find its likelihood, or to sample from the latent space and generate natural sounding output data. For TTS, NFs have just been introduced, yet there are already a number of high-quality systems and implementations available, such as: Flowtron [82], Glow-TTS [83], Flow-TTS [84], or Wave Tacotron [56]. From the generative perspective, this approach seems, at the moment, to be able to encompass all the desired goals of a speech synthesis system, but there are still a number of issues which need to be addressed, such as the inference time and latent space disentanglement and control.

All the above mentioned neural systems only solve the first part of the end-to-end problem, by taking the input text and converting it into a Mel spectrogram, or variations of it. For the spectrogram to be converted into an audio waveform, there is the separate component, called the vocoder. And there are also numerous studies on this topic dealing with the same trade-off issue of quality versus speed [85].

WaveNet [55] was one of the first neural networks designed to generate audio samples and achieved remarkably natural results. It is still the one vocoder to beat when designing new ones. However, its auto-regressive processes make it unfeasible for parallel inference, and several methods have been proposed to improve it, such as FFTNet [86] or Parallel WaveNet [87], but the quality is somewhat affected. Some other neural architectures used in vocoders are, of course, the recurrent networks used in WaveRNN [88] and LPCNet [89], or the adversarial architectures used in MelGAN [90], GELP [91], Parallel WaveGAN [92], VocGAN [93]. Following the trend of normalising flows-based acoustic modelling, flow-based vocoders have also been implemented. Some of the most remarkable being: FlowWaveNet [94], WaveGlow [95], WaveFlow [96], WG-WaveNet [97], EWG (Efficient WaveGlow) [98], MelGlow [99], or SqueezeWave [100].

In light of all these methods available for neural speech synthesis, it is again important to note the trade-offs between the quality of output speech, model sizes, training times, inference speed, computing power requirements and ease of control and adaptability. In the ideal scenario, a TTS system would be able to generate natural speech, at an order of magnitude faster than real-time processing speed, on a limited resource device. However, this goal has not yet been achieved by the current state-of-the-art, and any developer looking into TTS solutions should first determine the exact applicability scenario before implementing any of the above methods. It may be the case that, for example, in a limited vocabulary, non-interactive assistant, a simple formant synthesis system implemented on a dedicated hardware might be more reliable and adequate.

Some aspects which we did not take into account in the above enumeration are the multispeaker, multilingual TTS systems. However, in a commercial setup these are not directly required and can be substituted by independent high-quality systems integrated in a seamless way withing the IVA.

5. Open resources and tools

Deploying any research result into a commercial environment requires at least a baseline functional proof-of-concept from which to start optimising and adapting the system. It is the same in TTS systems, where especially the speech resources, text-processing tools, and system architectures can be at first tested and only then developed and migrated to the live solution. To aid this development, the following table indexes some of the most important resources and tools available for text to speech synthesis systems. This is by no means an exhaustive list, but rather a starting point. The official implementations pertaining to the published studies are marked as such. If no official implementation was found, we relied on our experience and prior work to link an open tool which comes as close as possible to the original publication.

Speech and text datasets and resources
Language Data Consortium (LDC) is a repository and distribution point for various language resources. Link: www.ldc.upenn.edu

The European Language Resources Association (ELRA) is a non-profit organisation whose main mission is to make Language Resources for Human Language Technologies available to the community at large. Link: www.elra.info/en/
META-SHARE [101] is an open and secure network of repositories for sharing and exchanging language data, tools and related web services. Link: www.meta-share.org
OpenSLR is a site devoted to hosting speech and language resources, such as training corpora for speech recognition, and software related mainly to speech recognition. Link: www.openslr.org
LibriVox is a group of worldwide volunteers who read and record public domain texts creating free public domain audiobooks for download. Link: www.librivox.org
Mozilla Common Voice is part of Mozilla's initiative to help teach machines how real people speak. Link: www.commonvoice.mozilla.org/en/datasets
Project Gutenberg is an online library of free eBooks. Link: www.gutenberg.org
LibriTTS [102] is a multi-speaker English corpus of approximately 585 hours of read English speech designed for TTS research. Link: www.openslr.org/60/
The Centre for Speech Technology Voice Cloning Toolkit (VCTK) Corpus includes speech data uttered by 109 native speakers of English with various accents. Each speaker reads out about 400 sentences. Link: www.datashare.is.ed.ac.uk/handle/10283/2950
CMU Wilderness Multilingual Speech Dataset [103] is a speech dataset of aligned sentences and audio for some 700 different languages. It is based on readings of the New Testament. Link: www.github.com/festvox/datasets-CMU_Wilderness
Text processing tools
Festival is a complete TTS system, but it enables the use of its front-end tools independently. It supports several languages and dialects. Link: www.cstr.ed.ac.uk/projects/festival/
CMUSphinx G2P tool is a grapheme-to-phoneme conversion tool based on transformers. Link: www.github.com/cmusphinx/g2p-seq2seq
Multilingual G2P uses the eSpeak tool to generate phonetic transcriptions in multiple languages. Link: www.github.com/jcsilva/multilingual-g2p.
Stanford NLP tools includes various text-processing and knowledge extraction tools for English and other languages. Link: www.nlp.stanford.edu/software/
RecoAPy [104] tool includes an easy to use interface for recording prompted speech, but also a set of models able to perform high accuracy phonetic transcription in 8 languages. Link: www.gitlab.utcluj.ro/sadriana/recoapy
word2vec [15] is a word embedding model that learns vector representations of words that capture semantic and other properties of these words from large amounts of text data. Link: code.google.com/archive/p/word2vec/
GloVE [16] is a word embedding method that learns from the co-occurences of words in text corpus obtaining similar vector representations for words that occur in the same context. Link: www.nlp.stanford.edu/projects/glove/
ELMo [17] obtains contextualized word embeddings that model the semantics and syntax of the word, but can learn different representations for various contexts. Link: www.allennlp.org/elmo
BERT [18] is a Transformer-based model that obtains context dependent word embeddings and can process sentences in parallel. Link: www.github.com/google-research/bert
Speech synthesis systems
eSpeak is a formant-based compact open source software speech synthesiser. Link: www.espeak.sourceforge.net/ [Official]
Festival is an unrestricted commercial and non-commercial use framework for building concatenative and HMM-based TTS systems. Link: www.cstr.ed.ac.uk/projects/festival/ [Official]
MaryTTS [105] is an open-source, multilingual TTS platform written in Java supporting diphone and unit selection synthesis. Link: http://mary.dfki.de/ [Official]

HTS [36] is the most commonly used implementation of the HMM-based speech synthesis. Link: http://hts.sp.nitech.ac.jp/ [Official]

Merlin [48] is a Python implementation of DNN models for statistical parametric speech synthesis. Link: www.github.com/CSTR-Edinburgh/merlin [Official]

IDLAK [106] is a project to build an end-to-end neural parametric TTS system within the Kaldi ASR framework. Link: www.idlak.readthedocs.io/en/latest/ [Official]

DeepVoice [50] follows the structure of HMM-based TTS systems, but replaces all its components with neural networks. Link: www.github.com/israelg99/deepvoice

Char2Wav [52] is an end-to-end neural model trained on characters that can synthesise speech with the SampleRNN vocoder. Link: https://github.com/sotelo/parrot [Official]

Tacotron [53] is one of the most frequently used end-to-end neural synthesis systems based on recurrent neural nets and attention mechanism. Link: www.github.com/keithito/tacotron

VoiceLoop [107] is one of the first neural synthesisers which uses a buffer memory instead of recurrent layers and does not require an audio-to-phone alignment. Link: www.github.com/facebookarchive/loop [Official]

Tacotron 2 [59] is an enhanced version of Tacotron which modifies the attention mechanism and also uses the WaveNet vocoder to generate the output speech. Link: www.github.com/NVIDIA/tacotron2

DeepVoice 3 [61] is a fully convolutional synthesis system that can synthesise speech in a multispeaker scenario. Link: www.github.com/r9y9/deepvoice3_pytorch

DCTTS [60] - Deep Convolutional TTS is a synthesis system that implements a two step synthesis, by first learning a coarse and then a fine-grained representation of the spectrogram. Link: www.github.com/tugstugi/pytorch-dc-tts

ClariNet [62] is the first text-to-wave neural architecture for speech synthesis, which is fully convolutional and enables fast end-to-end training from scratch. Link: www.github.com/ksw0306/ClariNet

Transformer TTS [67] replaces the recurrent structures of Tacotron 2 with attention mechanisms. Link: www.github.com/soobinseo/Transformer-TTS

GAN-TTS [79] is a GAN-based synthesis system that uses a generator to produce speech and multiple discriminators that evaluate the naturalness and text-adequacy of the output. Link: www.github.com/yanggeng1995/GAN-TTS

FastSpeech [68] is a novel feed-forward network based on Transformer which generates the Mel-spectrogram in parallel, and uses a teacher-based length predictor to achieve this parallel generation. Link: www.github.com/xcmyz/FastSpeech

FastSpeech 2 [69] is an enhanced version of FastSpeech where the length predictor teacher network is replaced by conditioning the output on duration, pitch and energy from extracted from the speech waveform at training and their predicted values in inference. Link: www.github.com/ming024/FastSpeech2

AlignTTS [70] is a feed-forward Transformer-based network with a duration predictor which aligns the speech and audio. Link: www.github.com/Deepest-Project/AlignTTS

Mellotron [108] is a multispeaker TTS able to emote emotions by explicitly conditioning on rhythm and continuous pitch contours from an audio signal. Link: www.github.com/NVIDIA/mellotron [Official]

Flowtron [82] is an autoregressive normalising flow-based generative network for TTS, also capable of transferring style from one speaker to another. Link: www.github.com/NVIDIA/flowtron [Official]

Glow-TTS [83] is a flow-based generative model for parallel TTS using a dynamic programming method to achieve the alignment between text and speech. Link: www.github.com/jaywalnut310/glow-tts [Official]

Speech synthesis system libraries

Mozilla TTS is a deep learning library for TTS that includes implementations for Tacotron, Tacotron 2, Glow-TTS and vocoders such as MelGAN, WaveRNN and others. Link: www.github.com/mozilla/TTS [Official]

NeMO is a toolkit that includes solutions for TTS, speech recognition and natural language processing tools as well. Link: www.github.com/NVIDIA/NeMo [Official]

ESPNET-TTS [109] is a toolkit that contains implementations for TTS systems like Tacotron, Transformer TTS, FastSpeech and others. Link: www.github.com/espnet/espnet [Official]

Parakeet is a flexible, efficient and state-of-the-art text-to-speech toolkit for the open-source community. It includes many influential TTS models proposed by Baidu Research and other research groups. Link: www.github.com/PaddlePaddle/Parakeet [Official]

Neural Vocoders

WaveNet [55] is an autoregressive and probabilistic model used to generate raw audio. It can also be conditioned on text to produce the very natural output speech, but its complexity makes it very resource demanding. Link: www.github.com/r9y9/wavenet_vocoder

WaveRNN [88] is a recurrent neural network based vocoder that is able to generate audio faster then real time as a result of its compact architecture. Link: www.github.com/fatchord/WaveRNN

FFTNet [86], inspired by WaveNet also generates the waveform samples sequentially, with the current sample being conditioned on the previous ones, but simplifies its architecture and allows real-time synthesis. Link: www.github.com/syang1993/FFTNet

nv-WaveNet is an open-source implementation of several different single-kernel approaches to the WaveNet variant described by [50]. Link: www.github.com/NVIDIA/nv-wavenet [Official]

LPCNet [89] is a variant of WaveRNN that improves the waveform generation by combining the recurrent neural architecture with linear prediction coefficients. Link: www.github.com/mozilla/LPCNet [Official]

FloWaveNet [94] is a generative model based on flows that can sample audio in real time. Compared to Parallel WaveNet and ClariNet it only requires a training process that is single-staged. Link: www.github.com/ksw0306/FloWaveNet [Official]

Parallel WaveGAN [95] is a vocoder that uses adversarial training and provides fast and lightweight waveform generation. Link: www.github.com/kan-bayashi/ParallelWaveGAN

WaveGlow [95] vocoder borrows from Glow and WaveNet to generate raw audio from Mel spectrograms. It is a flow-based model implemented with a single network. Link: www.github.com/NVIDIA/waveglow [Official]

MelGAN [90] is a GAN-based vocoder that is able to generate coherent waveforms, the model is non-autoregressive and based on convolutional layers. Link: www.github.com/descriptinc/melgan-neurips [Official]

GELP [91] is a parallel neural vocoder utilising generative adversarial networks, and integrating a linear predictive synthesis filter into the model. Link: www.github.com/ljuvela/GELP

SqueezeWave [100] is a lightweight version of WaveGlow that can generate on-device speech output. Link: https://github.com/tianrengao/SqueezeWave [Official]

WaveFlow [96] is a flow-based model that includes WaveNet and WaveGlow as special cases and can synthesise audio faster than real-time. Link: www.github.com/L0SG/WaveFlow

VocGAN [93] is a GAN-based vocoder that can synthesise speech in real time even on a CPU. Link: www.github.com/rishikksh20/VocGAN

WG-WaveNet [97] is composed of a WaveGlow like flow-based model combined with WaveNet based postfilter that can synthesise speech without the need for a GPU. Link: www.github.com/BogiHsu/WG-WaveNet

Speech synthesis challenges

Blizzard Challenge is a yearly challenge in which teams develop TTS systems starting from more or less the same resources, and are jointly evaluated in a large-scale listening test. Link: http://www.festvox.org/blizzard/

Voice Cloning Challenge is a bi-annual challenge in which teams are asked to provide a high-quality solution for cloning the voice of a target speaker within the same language, or cross-lingual. The results are also evaluated in a large scale listening test. Link: http://www.vc-challenge.org/

6. Quality measurements

Although there are no objective measures which can perfectly predict the perceived naturalness of the synthetic output [110, 111], we still need to measure a TTS system's performance. The current approach to doing this is to use *listening tests*. In a listening test, a set of listeners, preferably a large number of native speakers of the target language, are asked to rate the synthetic output in several scenarios using either absolute or relative values. The common setup includes multiple synthesis systems and natural samples. The evaluation can be performed by presenting one or two samples at a time and the listeners rate it by using a Mean Opinion Score (MOS) scale going from 1 to 5, with 5 being the highest value. Or, more commonly used nowadays, in a MUSHRA [112] setup, in which multiple samples are presented the same time and the listeners are asked to order and rate them on a scale of 1 to 100. There is also a preference test setup in which the listeners are asked to choose between two samples according to their preference or adequacy of the rendered speech to the text or speaker identity. The most common evaluation criteria are:

naturalness listeners are asked to rank how close to natural speech is a sample of synthetic output perceived;

intelligibility listeners are asked to transcribe what they hear after playing the sample only once. The transcripts are then compared to the reference transcript and the word error rate is computed;

speaker similarity listeners are presented with a natural sample as reference and a synthetic or natural sample for evaluation. They are asked to rate how similar the identity of the evaluation sample is in comparison to the reference sample.

7. Conclusions and open problems

In this chapter we aimed to provide a high-level indexing of the available methods to generate the voice of an IVA, and to provide the reader with a clear, informed starting point for developing his/her own text-to-speech synthesis system. In the recent years there has been an increasing interest in this domain, especially in the context of vocal chat bots and content access. So that it would be next to impossible to index all the publications and available tools and resources. Yet, we consider that the provided knowledge and minimal scientific description of the TTS domain is sufficient to trigger the interest and application of these methods in the reader's commercial products. It should also be clear that there is still an important trade-off between the quality and the resource requirements of the synthetic voices, and that a very thorough analysis of the applications' specifications and intended use should guide the developer into making the right choice of technology.

We should also point out that, although the recent advancements achieve close to human speech quality, there are still a number of issues that need to be addressed before we can easily say that the topic of speech synthesis has been thoroughly solved. One of these issues is that of *adequate prosody*. When synthesising long paragraphs, or entire books, there is still a lack of variability in the output, and a subset of certain prosodic patterns reemerge. Also, the problem of correctly emphasising certain words, or word groups, such that the desired message is clearly and correctly transmitted is still an open issue for TTS. There is also the problem of mimicking spontaneous speech, where repetitions, elisions, filled pauses, breaks and so on convey the mental process and effort of developing the message and generating it as a spoken discourse.

In terms of speaker identity, the fast adaptation, and also cross-lingual adaptation are of great interest to the TTS community at this point. Being able to copy a person's speech characteristics using as little examples as possible is a daunting task, yet giant leaps have been taken with the NN-based learning. More so, transferring the identity of a person speaking in a language, to the identity of a synthesis system generating a different language is also open for solutions.

On the more far-fetched goals is that of *affective rendering*. If we were to interact with a complete synthetic persona, we would like it to be adaptable to our state of mind, and render compassionate and emphatic emotions in its discourse. Yet the automatic detection and generation of emotions is far from being solved.

Author details

Adriana Stan[1*†] and Beáta Lőrincz[1,2†]

1 Technical University of Cluj-Napoca, Cluj-Napoca, Romania

2 "Babeş-Bolyai" University, Cluj-Napoca, Romania

*Address all correspondence to: adriana.stan@com.utcluj.ro

† These authors contributed equally.

References

[1] J. Benesty, M. M. Sondhi, and Y. A. Huang, *Springer Handbook of Speech Processing*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2007.

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

[3] "Missing fundamental," en.wikipedia.org/wiki/ Missing fundamental, online; accessed 15-December-2020.

[4] S. King, "Speech Zone - Windowing," speech.zone/windowing/, online; accessed 15-December-2020.

[5] "Fourier analysis," en.wikipedia.org/wiki/Fourier analysis, online; accessed 15-December-2020.

[6] "Cepstrum," en.wikipedia.org/wiki/Cepstrum, online; accessed 15-December-2020.

[7] J. Li, Z. Wu, R. Li, P. Zhi, S. Yang, and H. Meng, "Knowledge-Based Linguistic Encoding for End-to-End Mandarin Text-to-Speech Synthesis," in *Proc.* Interspeech *2019*, 2019, pp. 4494–4498.

[8] M. Nutu, B. Lorincz, and A. Stan, "Deep Learning for Automatic Diacritics Restoration in Romanian," in *Proc. of IEEE 15th International Conference on Intelligent Computer Communication and Processing*, 09 2019, pp. 1–5.

[9] H. Zhang, R. Sproat, A. H. Ng, F. Stahlberg, X. Peng, K. Gorman, and B. Roark, "Neural Models of Text Normalization for Speech Applications," Computational Linguistics, vol. 45, no. 2, pp. 293–337, 2019.

[10] B. Bohnet, R. McDonald, G. Simões, D. Andor, E. Pitler, and J. Maynez, "Morphosyntactic tagging with a meta-BiLSTM model over context sensitive token encodings," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 2642–2652.

[11] A. Cutler, *Lexical Stress*. John Wiley & Sons, Ltd, 2005, ch. 11, pp. 264–289.

[12] S. Thomas, M. N. Rao, H. A. Murthy, and C. S. Ramalingam, "Natural sounding TTS based on syllable-like units," in *14th European Signal Processing Conference, EUSIPCO 2006, Florence, Italy, September 4–8*, 2006. IEEE, 2006, pp. 1–5.

[13] A. Sokolov, T. Rohlin, and A. Rastrow, "Neural Machine Translation for Multilingual Grapheme-to-Phoneme Conversion," in *Proc.* Interspeech *2019*, 2019, pp. 2065–2069.

[14] "W3C - Speech Synthesis Markup Language (SSML) Version 1.1," https://www.w3.org/TR/speech-synthesis11/, online; accessed 15-December-2020.

[15] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[16] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[17] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," *arXiv preprint arXiv:1802.05365*, 2018.

[18] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of

deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[19] X. Huang, A. Acero, H.-W. Hon, and R. Reddy, *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*, 1st ed. USA: Prentice Hall PTR, 2001.

[20] D. H. Klatt, "Software for a cascade/parallel formant synthesizer," *Journal of The Acoustical Society of America*, vol. 67, 1980.

[21] J. Allen, S. Hunnicut, and D. Klatt, *From Text to Speech: the MITalk System*. Cambridge University Press, 1987.

[22] C. Bickley, K. Stevens, and D. Williams, "A framework for synthesis of segments based on pseudoarticulatory parameters," pp. 211–220, 1997.

[23] K. Richmond, Z.-H. Ling, and J. Yamagishi, "The use of articulatory movement data in speech synthesis applications: An overview - application of articulatory movements using machine learning algorithms [invited review]," Acoustical Science and Technology, vol. 36, no. 6, pp. 467–477, 2015.

[24] D. Hill, "gnuspeech," www.gnu.org/software/gnuspeech/, online; accessed 15-December-2020.

[25] A. Black, P. Taylor, and R. Caley,*The Festival Speech Synthesis System*, University of Edinburgh, 1999.

[26] T. Lambert and A. P. Breen, "A database design for a TTS synthesis system using lexical diphones," in *Proceedings of Interspeech*, 2004.

[27] T. Saito, Y. Hashimoto, and M. Sakamoto, "High-quality speech synthesis using context-dependent syllabic units," in *Proceedings of the Acoustics, Speech, and Signal Processing, 1996. on Conference Proceedings., 1996 IEEE International Conference – Volume 01*, ser. ICASSP '96, 1996, pp. 381–384.

[28] J. Matoušek, Z. Hanzlíček, and D. Tihelka, "Hybrid syllable/triphone speech synthesis," in *Proceedings of the 9th European Conference on Speech Communication and Technology*, Lisbon, Portugal, 2005, pp. 2529–2532.

[29] O. Buza, "Contribut, ii la analizas, şi sinteza vorbirii din text pentru limba română," Ph.D. dissertation, Technical University of Cluj-Napoca, 2010.

[30] R. Stetson, *Motor Phonetics: A Study of Speech Movements in Action*. Oberlin College, 1951.

[31] A. Black and N. Campbell, "Optimising selection of units from speech database for concatenative synthesis," in *Proc. EUROSPEECH-95*, Sep. 1995, pp. 581–584.

[32] A. Hunt and A. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *Proc. of ICASSP*, May 1996, pp. 373–376.

[33] Y. Qian, F. K. Soong, and Z. Yan, "A unified trajectory tiling approach to high quality speech rendering," IEEE Transactions on Audio, Speech, and Language Processing, vol. 21, no. 2, pp. 280–290, 2013.

[34] A. Falaschi, M. Giustiniani, and M. Verola, "A hidden Markov model approach to speech synthesis," in *Proceedings of Eurospeech*, vol. 1989, 1989, pp. 2187–2190.

[35] S. King, "An introduction to statistical parametric speech synthesis," Sadhana, vol. 36, p. 837–852, 2011.

[36] H. Zen, T. Toda, M. Nakamura, and K. Tokuda, "Details of Nitech HMM-based speech synthesis system for the Blizzard Challenge 2005," IEICE Trans.

Inf. & Syst., vol. E90-D, no. 1, pp. 325–333, Jan. 2007.

[37] J. Yamagishi, B. Usabaev, S. King, O. Watts, J. Dines, J. Tian, R. Hu, Y. Guan, K. Oura, K. Tokuda, R. Karhila, and M. Kurimo, "Thousands of voices for HMM-based speech synthesis – analysis and application of TTS systems built on various ASR corpora," IEEE Transactions on Audio, Speech and Language Processing, vol. 18, no. 5, pp. 984–1004, July 2010.

[38] J. Lorenzo-Trueba, R. Barra-Chicote, R. San-Segundo, J. Ferreiros, J. Yamagishi, and J. M. Montero, "Emotion transplantation through adaptation in hmm-based speech synthesis," Computer Speech and Language, vol. 34, no. 1, pp. 292–307, 2015.

[39] H. Kawahara, I. Masuda-Katsuse, and A. Cheveigné, "Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based F0 extraction: possible role of a repetitive structure in sounds," Speech Communication, vol. 27, pp. 187–207, 1999.

[40] M. Morise, F. Yokomori, and K. Ozawa, "WORLD: A Vocoder-Based High-Quality Speech Synthesis System for Real-Time Applications," IEICE Transactions, vol. 99-D, pp. 1877–1884, 2016.

[41] Q. Hu, K. Richmond, J. Yamagishi, and J. Latorre, "An experimental comparison of multiple vocoder types," in *8th ISCA Workshop on Speech Synthesis*, Barcelona, Spain, August 2013, pp. 155–160.

[42] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," Neurocomputing: Foundations of Research, p. 15–27, 1988.

[43] M. G. Rahim and C. C. Goodyear, "Articulatory synthesis with the aid of a neural net," in *International Conference on Acoustics, Speech, and Signal Processing*, 1989, pp. 227–230 vol.1.

[44] G. E. Hinton, "Learning multiple layers of representation," Trends in Cognitive Sciences, vol. 11, no. 10, pp. 428–434, 2007.

[45] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," Neural Computation, vol. 18, no. 7, pp. 1527–1554, 2006.

[46] Z. Ling, L. Deng, and D. Yu, "Modeling Spectral Envelopes Using Restricted Boltzmann Machines and Deep Belief Networks for Statistical Parametric Speech Synthesis," IEEE Transactions on Audio, Speech, and Language Processing, vol. 21, no. 10, pp. 2129–2139, 2013.

[47] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 7962–7966.

[48] Z. Wu, O. Watts, and S. King, "Merlin: An open source neural network speech synthesis system." In *Speech Synthesis Workshop*, 2016, pp. 202–207.

[49] O. Watts, G. Henter, J. Fong, and C. Valentini-Botinhao, "Where do the improvements come from in sequence-to-sequence neural TTS?" in *Proc of the 10th ISCA Speech Synthesis Workshop*. International Speech Communication Association, Sep. 2019, pp. 217–222.

[50] S. O. Arik, M. Chrzanowski, A. Coates, G. Diamos, A. Gibiansky, Y. Kang, X. Li, J. Miller, A. Ng, J. Raiman, S. Sengupta, and M. Shoeybi, "Deep voice: Real-time neural text-to-speech," *arXiv preprint arXiv:1702.07825*, 2017.

[51] W. Wang, S. Xu, and B. Xu, "First Step Towards End-to-End Parametric TTS Synthesis: Generating Spectral

Parameters with Neural Attention," in *Interspeech 2016*, 2016, pp. 2243–2247.

[52] J. Sotelo, S. Mehri, K. Kumar, J. F. Santos, K. Kastner, A. C. Courville, and Y. Bengio, "Char2wav: End-to-end speech synthesis," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24–26, 2017, Workshop Track Proceedings*. OpenReview.net, 2017.

[53] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous, "Tacotron: Towards end-to-end speech synthesis," in *Proc. of Interspeech*, 2017.

[54] D. Griffin and J. Lim, "Signal estimation from modified short-time Fourier transform," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 32, no. 2, pp. 236–243, 1984.

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

[56] R. J. Weiss, R. Skerry-Ryan, E. Battenberg, S. Mariooryad, and D. P. Kingma, "Wave-tacotron: Spectrogram-free end-to-end text-to-speech synthesis," *arXiv preprint arXiv: 2011.03568*, 2020.

[57] A. Peiró-Lilja and M. Farrús, "Naturalness Enhancement with Linguistic Information in End-to-End TTS Using Unsupervised Parallel Encoding," in *Proc.* Interspeech *2020*, 2020, pp. 3994–3998.

[58] J. Taylor and K. Richmond, "Enhancing Sequence-to-Sequence Text-to-Speech with Morphology," in *Proc.* Interspeech *2020*, 2020, pp. 1738–1742.

[59] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan, R. A. Saurous, Y. Agiomvrgiannakis, and Y. Wu, "Natural TTS Synthesis by Conditioning Wavenet on MEL Spectrogram Predictions," in *2018 IEEE International Conference on Acoustics,* Speech and Signal Processing (ICASSP), 2018, pp. 4779–4783.

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

[61] W. Ping, K. Peng, A. Gibiansky, S. O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep voice 3: 2000-speaker neural text-to-speech," Proc. ICLR, pp. 214–217, 2018.

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

[63] K. Peng, W. Ping, Z. Song, and K. Zhao, "Non-autoregressive neural text-to-speech," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. Virtual: PMLR, 13–18 Jul 2020, pp. 7586–7598.

[64] C. Yu, H. Lu, N. Hu, M. Yu, C. Weng, K. Xu, P. Liu, D. Tuo, S. Kang, G. Lei, D. Su, and D. Yu, "DurIAN: Duration Informed Attention Network for Speech Synthesis," in *Proc.* Interspeech *2020*, 2020, pp. 2027–2031.

[65] J. Shen, Y. Jia, M. Chrzanowski, Y. Zhang, I. Elias, H. Zen, and Y. Wu, "Non-attentive tacotron: Robust and controllable neural tts synthesis including unsupervised duration modeling," 2020.

[66] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," Advances in neural information processing systems, vol. 30, pp. 5998–6008, 2017.

[67] N. Li, S. Liu, Y. Liu, S. Zhao, and M. Liu, "Neural speech synthesis with transformer network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 6706–6713.

[68] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech: Fast, robust and controllable text to speech," in *Advances in Neural Information Processing Systems*, 2019, pp. 3171–3180.

[69] Y. Ren, C. Hu, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech 2: Fast and high-quality end-to-end text-to-speech," *arXiv preprint arXiv: 2006.04558*, 2020.

[70] Z. Zeng, J. Wang, N. Cheng, T. Xia, and J. Xiao, "Aligntts: Efficient feed-forward text-to-speech system without explicit alignment," in *ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6714–6718.

[71] D. Lim, W. Jang, G. O, H. Park, B. Kim, and J. Yoon, "JDI-T: Jointly Trained Duration Informed Transformer for Text-To-Speech without Explicit Alignment," in *Proc.* Interspeech *2020*, 2020, pp. 4004–4008.

[72] M. Chen, X. Tan, Y. Ren, J. Xu, H. Sun, S. Zhao, and T. Qin, "Multispeech: Multi-speaker text to speech with transformer," *arXiv preprint arXiv: 2006.04664*, 2020.

[73] H. R. Ihm, J. Y. Lee, B. J. Choi, S. J. Cheon, and N. S. Kim, "Reformer-TTS: Neural Speech Synthesis with Reformer Network," Proc. Interspeech *2020*, pp. 2012–2016, 2020.

[74] W.-N. Hsu, Y. Zhang, and J. Glass, "Learning latent representations for speech generation and transformation," *arXiv preprint arXiv:1704.04222*, 2017.

[75] Y.-J. Zhang, S. Pan, L. He, and Z.-H. Ling, "Learning latent representations for style control and transfer in end-to-end speech synthesis," in *ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6945–6949.

[76] G. Sun, Y. Zhang, R. J. Weiss, Y. Cao, H. Zen, A. Rosenberg, B. Ramabhadran, and Y. Wu, "Generating diverse and natural text-to-speech samples using a quantized fine-grained VAE and autoregressive prosody prior," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6699–6703.

[77] Y. Yasuda, X. Wang, and J. Yamagishi, "End-to-End Text-to-Speech using Latent Duration based on VQ-VAE," *arXiv preprint arXiv: 2010.09602*, 2020.

[78] V. Aggarwal, M. Cotescu, N. Prateek, J. Lorenzo-Trueba, and R. Barra-Chicote, "Using VAEs and Normalizing Flows for One-Shot Text-To-Speech Synthesis of Expressive Speech," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6179–6183.

[79] M. Bińkowski, J. Donahue, S. Dieleman, A. Clark, E. Elsen, N. Casagrande, L. C. Cobo, and K. Simonyan, "High fidelity speech synthesis with adversarial networks," *arXiv preprint arXiv:1909.11646*, 2019.

[80] H. Guo, F. K. Soong, L. He, and L. Xie, "A new GAN-based end-to-end TTS training algorithm," *arXiv preprint arXiv:1904.04775*, 2019.

[81] D. J. Rezende and S. Mohamed, "Variational inference with normalizing

flows," *arXiv preprint arXiv:1505.05770*, 2015.

[82] R. Valle, K. Shih, R. Prenger, and B. Catanzaro, "Flowtron: an autoregressive flow-based generative network for text-to-speech synthesis," *arXiv preprint arXiv:2005.05957*, 2020.

[83] J. Kim, S. Kim, J. Kong, and S. Yoon, "Glow-TTS: A Generative Flow for Text-to-Speech via Monotonic Alignment Search," *arXiv preprint arXiv:2005.11129*, 2020.

[84] C. Miao, S. Liang, M. Chen, J. Ma, S. Wang, and J. Xiao, "Flow-TTS: A Non-Autoregressive Network for Text to Speech Based on Flow," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7209–7213.

[85] P. Govalkar, J. Fischer, F. Zalkow, and C. Dittmar, "A Comparison of Recent Neural Vocoders for Speech Signal Reconstruction," in *Proc. 10th ISCA Speech Synthesis Workshop*, 2019, pp. 7–12.

[86] Z. Jin, A. Finkelstein, G. J. Mysore, and J. Lu, "FFTNet: A Real-Time Speaker-Dependent Neural Vocoder," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 2251–2255.

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

[88] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. v. d. Oord, S. Dieleman, and K. Kavukcuoglu, "Efficient neural audio synthesis," *arXiv preprint arXiv:1802.08435*, 2018.

[89] J.-M. Valin and J. Skoglund, "LPCNet: Improving neural speech synthesis through linear prediction," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5891–5895.

[90] K. Kumar, R. Kumar, T. de Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. de Brébisson, Y. Bengio, and A. C. Courville, "MelGAN: Generative adversarial networks for conditional waveform synthesis," in *Advances in Neural Information Processing Systems*, 2019, pp. 14 910–14 921.

[91] L. Juvela, B. Bollepalli, J. Yamagishi, and P. Alku, "GELP: GAN-excited linear prediction for speech synthesis from mel-spectrogram," *arXiv preprint arXiv: 1904.03976*, 2019.

[92] 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 *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6199–6203.

[93] J. Yang, J. Lee, Y. Kim, H.-Y. Cho, and I. Kim, "VocGAN: A High-Fidelity Real-Time Vocoder with a Hierarchically-Nested Adversarial Network," in *Proc.* Interspeech *2020*, 2020, pp. 200–204.

[94] S. Kim, S.-g. Lee, J. Song, J. Kim, and S. Yoon, "FloWaveNet: A generative flow for raw audio," *arXiv preprint arXiv:1811.02155*, 2018.

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

[96] W. Ping, K. Peng, K. Zhao, and Z. Song, "WaveFlow: A compact flow-based model for raw audio," in *International Conference on Machine Learning*. PMLR, 2020, pp. 7706–7716.

[97] P. chun Hsu and H. yi Lee, "WG-WaveNet: Real-Time High-Fidelity Speech Synthesis Without GPU," in *Proc.* Interspeech, 2020, pp. 210–214.

[98] W. Song, G. Xu, Z. Zhang, C. Zhang, X. He, and B. Zhou, "Efficient WaveGlow: An Improved WaveGlow Vocoder with Enhanced Speed," in *Proc.* Interspeech, 2020, pp. 225–229.

[99] Z. Zeng, J. Wang, N. Cheng, and J. Xiao, "MelGlow: Efficient Waveform Generative Network Based on Location-Variable Convolution," *arXiv preprint arXiv:2012.01684*, 2020.

[100] B. Zhai, T. Gao, F. Xue, D. Rothchild, B. Wu, J. E. Gonzalez, and K. Keutzer, "SqueezeWave: Extremely Lightweight Vocoders for On-device Speech Synthesis," *arXiv preprint arXiv: 2001.05685*, 2020.

[101] M. Gavrilidou, P. Labropoulou, E. Desipri, S. Piperidis, H. Papageorgiou, M. Monachini, F. Frontini, T. Declerck, G. Francopoulo, V. Arranz, and V. Mapelli, "The META-SHARE Metadata Schema for the Description of Language Resources." in *LREC*, 2012, pp. 1090–1097.

[102] H. Zen, V. Dang, R. Clark, Y. Zhang, R. J. Weiss, Y. Jia, Z. Chen, and Y. Wu, "LibriTTS: A corpus derived from LibriSpeech for text-to-speech," *arXiv preprint arXiv:1904.02882*, 2019.

[103] A. W. Black, "CMU Wilderness Multilingual Speech Dataset," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 5971–5975.

[104] A. Stan, "Recoapy: Data recording, pre-processing and phonetic transcription for end-to-end speech-based applications," *arXiv preprint arXiv:2009.05493*, 2020.

[105] M. Schröder, M. Charfuelan, S. Pammi, and I. Steiner, "Open source voice creation toolkit for the MARY TTS Platform," in *Proc. of Interspeech*, 2011.

[106] B. Potard, M. P. Aylett, D. A. Baude, and P. Motlicek, "Idlak Tangle: An Open Source Kaldi Based Parametric Speech Synthesiser Based on DNN." in *Proc. of Interspeech*, 2016, pp. 2293–2297.

[107] Y. Taigman, L. Wolf, A. Polyak, and E. Nachmani, "Voiceloop: Voice fitting and synthesis via a phonological loop," *arXiv preprint arXiv:1707.06588*, 2017.

[108] R. Valle, J. Li, R. Prenger, and B. Catanzaro, "Mellotron: Multispeaker expressive voice synthesis by conditioning on rhythm, pitch and global style tokens," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6189–6193.

[109] T. Hayashi, R. Yamamoto, K. Inoue, T. Yoshimura, S. Watanabe, T. Toda, K. Takeda, Y. Zhang, and X. Tan, "ESPnet-TTS: Unified, reproducible, and integratable open source end-to-end text-to-speech toolkit," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7654–7658.

[110] Y. Choi, Y. Jung, and H. Kim, "Deep MOS Predictor for Synthetic Speech Using Cluster-Based Modeling," in *Proc.* Interspeech *2020*, 2020, pp. 1743–1747.

[111] M. Wester, C. Valentini-Botinhao, and G. E. Henter, "Are we using enough listeners? No! An empirically-supported critique of Interspeech 2014 TTS evaluations," in *Proc. Interspeech*, Dresden, September 2015, pp. 3476–3480.

[112] M. Schoeffler, S. Bartoschek, F.-R. Stöter, M. Roess, S. Westphal, B. Edler, and J. Herre, "webMUSHRA — A Comprehensive Framework for Web-based Listening Tests," *Journal of Open Research Software*, vol. 6, no. 1, p. 8, Feb. 2018, number: 1 Publisher: Ubiquity Press.