Ludwig-Maximilians-Universität München

WiSe 2021/22

Conversational AI Seminar

Text-To-Speech Synthesis

April 24, 2022

Author: Laura Isla Navarro

MSc Computerlinguistik mit Nebenfach Informatik

Contents

1 Introduction

The high availability of technological resources within society's reach has driven the need for applications that can enhance the user's accessibility to these artifacts. The existence of audio-based content is especially important for the blind, who can get text read out loud, or sufferers of neurological disorders such as Stephen Hawking, who spoke by manipulating a Text-To-Speech (TTS) system.

Despite being one of the earliest goals of Natural Language Processing, speech synthesis tasks such as Automatic Speech Recognition and TTS synthesis have been researched since long before computers existed. As a matter of fact, the Austrian-Hungarian inventor Wolfgang von Kempelen built between 1769 and 1790 a device that emulated the workings of the human speech system and could deliver consonant and vowel sounds when operated [1]. Nowadays, the TTS systems available in the market are not made of wood and leather. Instead, they usually are neural architectures composed by an encoder and decoder especially designed for technological devices. TTS architectures have also evolved over time from parametric to deep learning methods. Each of these will be briefly discussed on this paper.

It is also worth mentioning that recent research is working on the task of multi-speaker TTS, a key dependency for the growth of TTS systems [2], and tailor them with the user's preferences in mind.

In the following sections, we will introduce the steps followed by a TTS system to obtain the desired output, a waveform representing speech sound, as well as an overview into the evolution of TTS systems. Next, despite there being many neural TTS synthesizers available, we will set the focus on the end-to-end TTS synthesizer Tacotron 2. Lastly, we will discuss the evaluation metrics that are used to assess TTS systems.

2 TTS Pipeline

Essentially, TTS systems consist of a front-end and a back-end component. The latter turns a raw text into their equivalent written-out words. Once this normalization process has occurred, each word is mapped to its phonetic transcription. Furthermore, the front-end engine splits the text into prosodic units generating an output of mel spectrograms (Figure 1). The back-end system or synthesizer will then convert this output into waveforms via a vocoder.

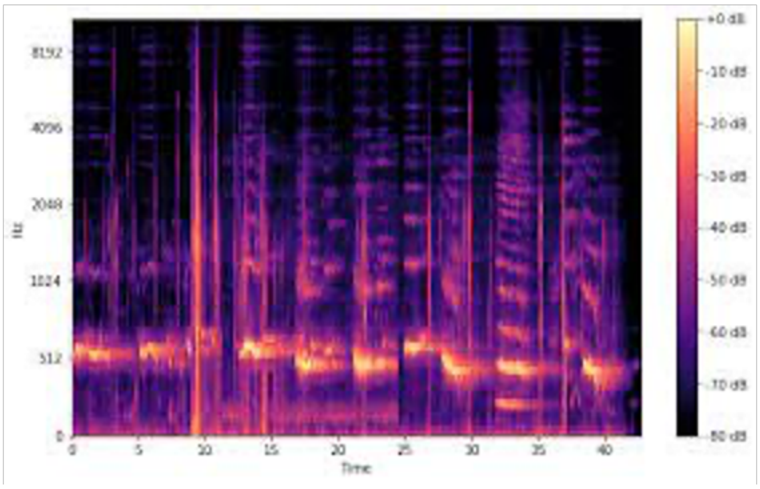

Figure 1: Mel-spectrogram. The x-axis represents the time, and the y-axis the frequency.

2.1 Input normalization

Since text may contain non-standard words such as numbers, monetary amounts, and dates, among others, which are pronounced differently than they are spelled, it is paramount to normalize these inputs. In some languages like English, the verbalization of a non-standard word depends on its semiotic class [3]. In others with grammatical gender like French, the normalization process takes morphological features into account. In German, the grammatical case of the noun plays a role [4].

While end-to-end TTS systems may be able to normalize themselves, their amount of training data is too limited to produce accurate outputs when it comes to non-standard words, making the creation of a separate normalization step an intrinsic part of these systems.

Alongside these normalization techniques, words are labelled by their part of speech, and converted to a phonetic representation before becoming a high-level representation of audio.

2.1.1 Rule-based normalization

When designing a normalizer, one can opt for a rule-based or an encoder-decoder model. In rule-based models, the input is first tokenized, that is, it passes through a series of hand-written write rules in the form of regular expressions to detect non-standard words. Secondly, the rules in the verbalization stage will determine how to verbalize each semiotic class. The more complex Kestrel text normalization system [5] is used in larger TTS applications. The peculiarity of this normalization system is that it separates the initial tokenization and classification step from the verbalization phase. When the semiotic classes at hand are detected in the grammars compiled into weighted finite-state transducers (WFSTs), they pass to the verbalization step. Figure 2 shows how this system works.

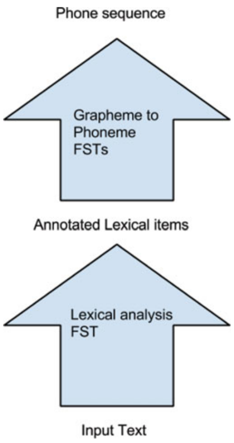

Figure 2: Pipeline of the rule-based normalization strategy. Figure from [5]

Although rule-based models can be advantageous, since they do not require training data, these rules can be weak and would need expert human rule-writers, which results in a highly costly and hard-to-maintain model.

2.1.2 Encoder-decoder-based normalization

Encoder-decoder-based normalization methods are presented as an alternative to the problematic rule-based normalization techniques – hand-written language-specific grammars, for state-of-the-art industrial systems [6]. The neural networks that conform encoder-decoder models treat text normalization as sequence-to-sequence problem as a machine translation task may be. The training sets applied to replace non-standard words with their corresponding verbalization are expert-labeled. Based on this, there is a system trained to map from an input including the non-standard words to an output where the non-standard words have been written out as they should be verbalized.

Although rare, one of the biggest issues concerning these systems is that these models commit some inappropriate mistakes like verbalizing 3 cm as three kilometers. Therefore, Zhang et al. [6] developed a series of finite-state covering grammars learned from data that should solve this. This should help the neural networks during training and decoding, or just during decoding.

2.2 Spectrogram prediction

Before the already normalized input turns into sound, there are a series of processes taking place to map character embeddings to their mel-scale spectrograms. It is noteworthy that a successful speech synthesis system aims at delivering an audio output which is both natural and intelligible to human audition. With the rapid development of the computer science industry, there are a plethora of approaches to achieve this objective, which will be discussed in the following section.

2.2.1 Speech synthesis approaches

In the early days of speech synthesizer technologies, concatenation and parametric synthesis used to be widely applied. However, with the rise of deep learning methods, the focus has been shifted from human-engineered speech features to fully machine-obtained parameters.

2.2.1.1 Concatenative Text-to-Speech Synthesis This approach implies the stringing of segments of recorded speech units, which subsequently form an ample database. These speech units must be pre-recorded from a single speaker before the synthesis occurs. Whereas the quality of the audio will be highly intelligible, concatenative approaches do not consider prosody resulting in audible glitches in the output.

2.2.1.2 Formant Text-to-Speech Synthesis Based on a set of specified rules, formant synthesizers generate a series of artificial signals, which mimic spectral properties of natural speech. In contrast with concatenative systems, formant-based cannot only deliver highly intelligible speech even at high speeds, but also can leverage all features of the output speech, generating a wide range of emotions.

2.2.1.3 Parametric Text-to-Speech Synthesis Since concatenative methods do not address prosody that effectively, it was necessary to develop more statistical approaches. In parametric synthesis, a series of parameters such as fundamental frequency, and magnitude spectrum are combined to deliver all sorts of speech. Firstly, linguistic features such as phonemes are extracted from the textual input and are fed to the vocoder along with so-called vocoder features, which are conformed by spectrograms, and fundamental frequencies, among others. These vocoder features represent some inherent property of human speech, and are needed when processing audio.

Using Hidden Semi-Markov models, the vocoder generates a waveform, and estimates speech parameters such as speech rate, and intonation.

2.2.1.4 Deep Learning-based Speech Synthesis The caveat of HMM-based techniques is that their linguistic features are hardcoded by humans, which might not be the best features to synthesize speech [8]. Furthermore, the mapping of these acoustic features into probability densities of speech parameters, using several decision trees, can result in muffled speech due to the oversmoothing of these features [9]. This sort of synthesis could be represented as follows, where Y is an input text sequence, X the target speech to be derived, and the model's parameters:

$$X = arg\,max P(X|Y,\theta)$$

Figure 3 offers a high-level representation of the pipeline for a neural TTS system.

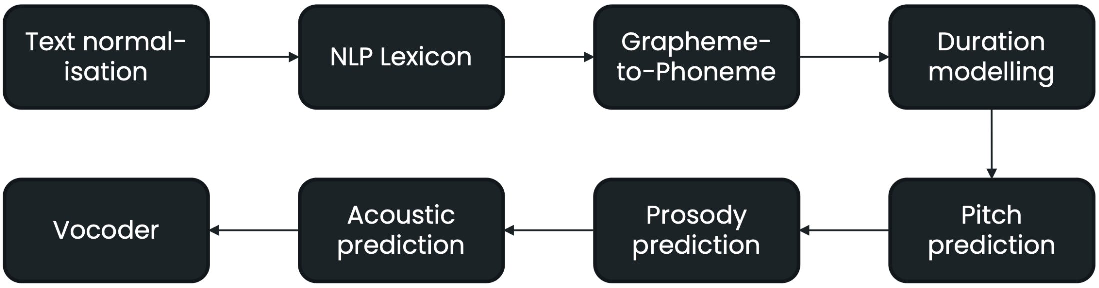

Figure 3: Pipeline of a neural TTS system.

2.3 Vocoder

The vocoding process aims at turning an intermediate form of audio into an audible waveform. This is a critical step in TTS synthesis since the choice of vocoder determines the final audio quality. Traditionally, vocoders accomplished this task applying digital signal processing techniques. Nowadays, with the crescent use of neural methods, this is done through neural networks comprising encoding and decoding processes.

2.3.1 A neural vocoder: WaveNet

WaveNet [10] is an autoregressive network where each sample depends on the previous on the prior ones. Mathematically, the probability of a sequence $Y = y_1, \ldots, y_t$ given an intermediate input mel-spectrogram h is computed as:

$$p\theta(x) = \prod_{t=1}^{T} p(x_t|x_1, ..., x_t - 1)$$

This probability distribution is made up of a series of dilated convolutions, which are a subtype of causal convolutional layer, and which solely focus on the previous input. Figure 4 sketches the computation of the output at time t with four dilated convolutional layers.

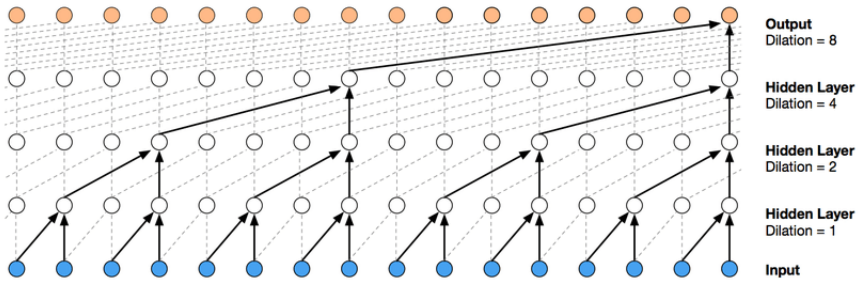

Figure 4: Dilated convolutional layers. Figure from [10]

In the following section, we will set the focus on Tacotron 2, an end-to-end DNN-based speech synthesizer, which joins an encoder-decoder architecture, the Tacotron system, with a vocoder, WaveNet, to produce more human-like outputs.

2.4 A case in point: Tacotron 2

The release of DeepMind's WaveNet neural vocoder in 2016 marked a turning point in TTS research. WaveNet is not a complete TTS synthesis solution on its own, since it lacks a synthesizer able to convert phonemes into high-level representation of audio. In 2017, Shen et al. [11] presented Tacotron 2, a synthesizer made of a sequence-to-sequence network, which maps character embeddings to their mel-scale spectrograms, which are then processed by the WaveNet vocoder. This approach delivers more realistic than the already existing Tacotron (Figure 5) synthesizer [12], which employed a vocoder based on the Griffin-Lim algorithm for waveform synthesis.

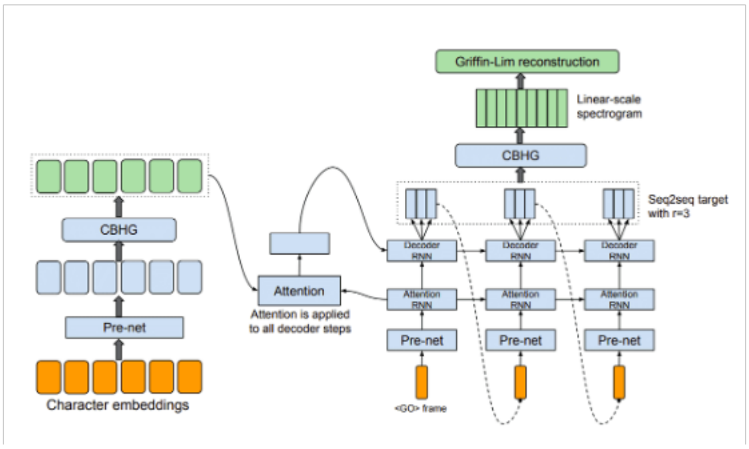

Figure 5: Encoder-decoder of the Tacotron architecture followed by the Griffin-Lim algorithm for the vocoder part. Figure from [12]

Essentially, Tacotron 2 (Figure 6) enhances and simplifies the original architecture of the autoregressive model Tacotron. The core part of the Tacotron architecture is a CBHG module (Figure 7), composed by a 1-D convolution bank, highway network, and a bidirectional GRU. Originally, this CBHG was meant for machine translation tasks, but it is also useful for speech synthesis due to the sequentiality of speech data. However, in Tacotron 2 this module is replaced with 3 convolutional layers and a LSTM. Furthermore, Tacotron 2 adopts location sensitive attention, which extends the additive attention mechanism applied in Tacotron and applies a series of mel-filters which deliver mel-spectrograms, more sophisticated than the linear-log spectrograms of Tacotron. This encourages that the model moves forward consistently, avoiding that decoder repeats or ignores certain subsequences [13].

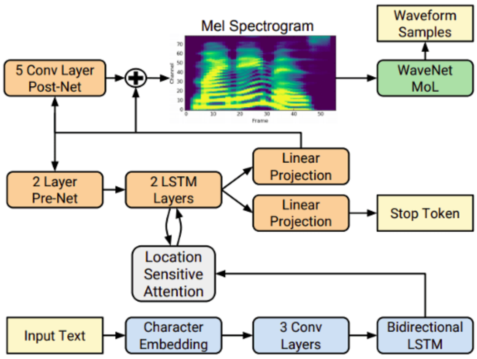

Figure 6: Encoder-decoder architecture of Tacotron 2. The location-sensitive attention system is the meeting point for the encoder and decoder. Figure from [11]

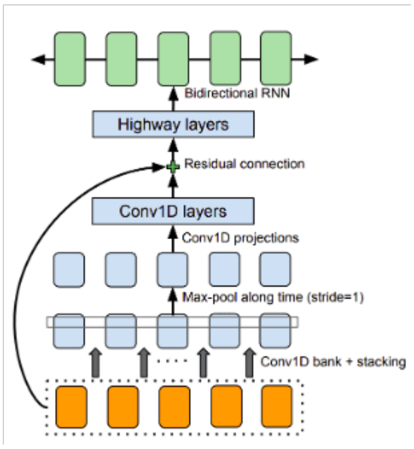

Figure 7: CBHG module. Part of the decoder in Tacotron. Figure from [12]

The decoder in Tacotron 2 is an autoregressive LSTM, which generates one slice of the mel-spectrogram at a time based on the encoded input. The predicted mel-spectrum from the previous time slot passes through a two-layered pre-net containing two fully connected layers of 256 hidden ReLU units. Afterwards, it is concatenated with the vector context resulting from the attention mechanism before being processed by two unidirectional LSTM layers with 1024 units. Next, the mel-spectrogram passes through a linear layer connected to a 5-layered convolutional post-net, which predict one 80-dimensional log-mel filterbank vector frame. Subsequently, a sigmoidal linear layer is put in place to decide whether to stop producing output [1].

3 Evaluation metrics

One of the most challenging parts in the TTS process is the evaluation of the actual system. TTS model evaluation relies on two types of tests: subjective tests and objective tests [14]. The latter require humans to assess the speech quality and naturalness, whereas the objective ones apply speech signal processing algorithms which measure voice performance. According to Jurafsky and Martin [1], intelligibility and quality are the metrics to be estimated by a TTS evaluation system.

The Mean Opinion Score (MOS) (Table 1) belongs to the subjective category and is widely used to measure the quality of the generated speech by a particular TTS system. On a scale from 1 to 5, from bad quality to excellent quality, users are asked to rate how good the outputs are. Human utterances usually score between 4.3 and 4.5, whereas outputs obtaining 3.5 or less are considered mediocre. T-tests accompany MOS to test for significant differences[1]. Alternatively, if there are two TTS systems to be compared to check whether a feature change has made a qualitative difference in the synthesized output, AB subjective tests (Table 2) are put into place. Furthermore, there are further subjective tests to measure intelligibility such as the Modified Rhyme Test (MRT) or the Semantically Unpredictable Sentence (SUS). On the objective side, the Mel Cepstral Distortion (MCD) test, and the perceptual evaluation of speech quality (PESQ) help to assess TTS systems algorithmically.

The need for human evaluators makes this task rather expensive and time-consuming, therefore, the development of an alternative automatic metric to evaluate TTS synthesizers remains open.

[1] The interest reader can go to this website https://google.github.io/tacotron/publications/tacotron2/ to obtain an idea of how Tacotron compares to Tacotron qualitative speaking. Table 3 also displays the MOS scores for the aforementioned systems, as well as for the linguistic WaveNet, and the ground truth.

Rating	Speech Quality	Level of Distortion
5	Excellent	Imperceptible
4	Good	Just perceptible, but not annoying
3	Fair	Perceptible and slightly annoying
2	Poor	Annoying, but not objectionable
1	Bad	Very annoying and objectionable

Table 1: MOS rating scale. Table from [14]

Rating	Quality of Synthesized Audios by Systems A and B
3	A very good
2	A better
1	A good
0	About the same
1	B good
2	B better
3	B very good

Table 2: AB rating scale. Table from [14]

System	MOS
Tacotron	4.001 ± 0.087
Linguistic WaveNet	4.341 ± 0.051
Ground Truth	4.582 ± 0.053
Tacotron 2	4.526 ± 0.066

Table 3: MOS scores. Table from [11]

4 Conclusion

The goal of this seminar paper was to offer an overview of the complex TTS synthesis world. In the past, parametric and formant models were popular choices when creating a TTS system, but as the computational power of GPUs and CPUs increase, so does the availability of neural-based approaches. This also results in more human-like outputs, which greatly enhances user experience in commercial applications of TTS synthesizers.

Despite the quality improvement of TTS technologies, as seen in the closeness of the Tacotron 2 MOS score to that of the ground truth on table 3, TTS synthesis still has a long way to go, especially when it comes to multilingual speech synthesis, since most of the current systems are made for high-resource languages. Furthermore, the way TTS systems are evaluated needs to be further researched, to make the process cheaper and more efficient.

References

[1] Dan Jurafsky and James H. Martin. Speech and language processing. chapter 26. Prentice Hall, 2021. ISBN 0135041961.

[2] Julia Cambre, Jessica Colnago, Jim Maddock, Janice Tsai, and Jofish Kaye. Choice of Voices: A Large-Scale Evaluation of Text-to-Speech Voice Quality for Long-Form Content. *CHI Conference on Human Factors in Computing Systems (CHI '20)*, page 14 Pages, 2020.

[3] Paul Taylor. *Text-to-Speech Synthesis*. Cambridge University Press, 2009.

[4] Vera Demberg. Letter-to-phoneme conversion for a German Text-To-Speech system. Master's thesis, Universität Stuttgart, 2006.

[5] Peter Ebden and Richard Sproat. The Kestrel TTS text normalization system. *Natural Language Engineering*, 21(3):333, 2015.

[6] Hao Zhang, Richard Sproat, Axel H. Ng, Felix Stahlberg, Xiaochang Peng, Kyle Gorman, and Brian Roark. Neural models of text normalization for speech applications. *Computational Linguistics*, 45(2):293–337, 2019.

[7] Arsenii Dunaev. A Text-To-Speech system based on Deep Neural Networks, 2019.

[8] Utkarsh Saxena. Speech synthesis techniques using deep neural networks. https://medium.com/@saxenauts/speech-synthesis-techniques-using-deep-neural-networks-38699e943861, 2017.

[9] Yishuang Ning, Sheng He, Zhiyong Wu, Chunxiao Xing, and Liang-Jie Zhang. A review of deep learning based speech synthesis. *Applied Sciences*, 2019.

[10] Aaron van den Oord, Yazhe Li, Igor Babuschkin, Karen Simonyan, Oriol Vinyals, Koray Kavukcuoglu, George van den Driessche, Edward Lockhart, Luis C. Cobo, Florian Stimberg, Norman Casagrande, Dominik Grewe, Seb Noury, Sander Dieleman, Erich Elsen, Nal Kalchbrenner, Heiga Zen, Alex

Graves, Helen King, Tom Walters, Dan Belov, and Demis Hassabis. Parallel wavenet: Fast high-fidelity speech synthesis, 2018. URL https://arxiv.org/abs/1711.10433.

[11] Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4779–4783. IEEE, 2018.

[12] Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, Quoc Le, Yannis Agiomyrgiannakis, Rob Clark, and Rif A. Saurous. Tacotron: Towards end-to-end speech synthesis, 2017. URL https://arxiv.org/abs/1703.10135.

[13] Jan Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. Attention-based models for speech recognition. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 577–585, Cambridge, MA, USA, 2015. MIT Press.

[14] A. Valizada, S. Jafarova, E. Sultanov, and S Rustamov. Development and evaluation of speech synthesis system based on deep learning models. *Symmetry*, 13, 2021. doi: https://doi.org/10.3390/sym13050819.

[15] Matrapazis Anastasios. Greek Text-to-Speech. Master's thesis, Athens University of Economics and Business, 2021.

Selbstständigkeitserklärung

Hiermit versichere ich, dass diese Arbeit von mir persönlich verfasst ist und dass ich keinerlei fremde Hilfe in Anspruch genommen habe. Ebenso versichere ich, dass diese Arbeit oder Teile daraus weder von mir selbst noch von anderen als Leistungsnachweise andernorts eingereicht wurden. Wörtliche oder sinngemäße Übernahmen aus anderen Schriften und Veröffentlichungen in gedruckter oder elektronischer Form sind gekennzeichnet. Sämtliche Sekundärliteratur und sonstige Quellen sind nachgewiesen und in der Bibliographie aufgeführt. Das Gleiche gilt für graphische Darstellungen und Bilder sowie für alle Internet-Quellen. Ich bin ferner damit einverstanden, dass meine Arbeit zum Zwecke eines Plagiatsabgleichs in elektronischer Form anonymisiert versendet und gespeichert werden kann. Mir ist bekannt, dass von der Korrektur der Arbeit abgesehen und die Prüfungsleistung mit nicht ausreichend bewertet werden kann, wenn die Erklärung nicht erteilt wird.