Data-Driven Waveform Generation for High-Quality Text-to-Speech (TTS) Synthesis

Informatics Research Review

Emily Balloun – emilyballoun@gmail.com – s1246171

Submitted January 17th, 2013

Contents

1 Introduction – Background and Contribution

1.1 Background

The output synthetic voice of a text-to-speech (TTS) system is typically modeled after the natural voice of a single speaker. Due to the increases in computational power over the last decades, advances in methods for creating synthetic voices have become data-driven (aka corpus-based) rather than based on expert heuristics [1], [2]. The intelligibility and naturalness of a voice determine its level of quality, and while most voices are intelligible, because of the complexity of speech synthesis for natural language, the issue of naturalness in synthesized speech has not yet been completely solved. The two predominant methods, unit selection synthesis and hidden Markov model (HMM) -based synthesis will be considered here in terms of their ability to generate high-quality natural and intelligible voices for TTS systems. One major reason that TTS systems are not more widely adopted is that users generally reject voices which sound unnatural [3].

The most natural-sounding and intelligible commercial synthetic voices are typically created by using many hours of phonetically-normalized recordings which are processed for unit selection [1]. However, these recordings are expensive and limit the number of voices it is feasible to make [4], and the size, or footprint, of the unit selection database combined with the cost of making voices limits the number of applications they can be used for. Despite this, unit selection synthesis is the de facto industry standard at present.

If natural-sounding and intelligible voices could be made using less costly processes, then more voices for video games, screen readers, translation systems and other systems could be created, and it could be even become possible to make voices for individual users quickly and easily; these could be used for a variety of personal tasks such as reading texts, or as components of assistive speech devices. A source of current interest in TTS is centered on the mobile phone market, with an increasing demand for applications that can be used on smartphones and other consumer devices.

1.2 Contribution

This research review examines the evolution of current industry-standard and state-of-the-art techniques in waveform generation for TTS systems to provide a comparison of unit selection synthesis to HMM-based synthesis in terms of their respective potential in research and industry. This review will also briefly introduce currently-trending hybrid unit selection and HMM approaches that aim to capture the best of both techniques.

As a point of reference, Section 2 offers an overview of TTS synthesis and its quality goals, and the evaluation metrics used to measure system quality against these goals. Section 3 broadly covers unit selection synthesis, outlining a typical unit selection waveform generation system and the advantages and disadvantages of such a system. Section 4 examines the evolution of HMM-based statistical parametric synthesis, highlighting the main challenges and potentials of such synthesis. Section 5 introduces hybrid systems that attempt to capitalize on the best of both types of popular synthesis. Section 6, compares the three methods of waveform generation described in Sections 3, 4, and 5 and illustrates their scale and impact, now and for the future. To close, Section 7 looks at some directions and challenges remaining in the field of waveform generation for TTS.

2 TTS Quality Goals and Evaluation

2.1 TTS quality goals

TTS synthesis is the process of taking some text input through processing and analysis that produces a linguistic specification of the given text which includes a sequence of phones, durations, and sometimes prosodic information which acts as a recipe for waveform generation. This linguistic specification is the instructions for waveform generation based on analysis of text, and waveform generation is the conversion of a linguistic specification to an acoustic speech waveform [5, p. 284]. The broad goal of TTS systems is that the best output (the minimum error) of a speech synthesizer should be indistinguishable from a human voice, as reflected in Equation (1).

$$\forall_x SyntheticVoice \rightarrow minError(x) = NaturalVoice \tag{1}$$

2.2 Evaluation

Overall quality of synthesis from a waveform generation system can be evaluated against two criteria. First, the output must be intelligible by humans. Secondly, the output should sound like natural human speech, based on prosodic and phonetic quality. Objective evaluation is possible through using a variety of calculable measures of distance, often from a gold standard recording of natural speech. However, objective evaluation is often less reliable than subjective measures, in which human listeners evaluate speech [4]. Objective measures are not included here.

Subjective Evaluation: Intelligibility The most common subjective evaluations of intelligibility are dictation tasks, tasks in which human interlocutors transcribe the perceived output of a TTS system [4], [5, p. 315]. Stimuli for dictation tasks are typically printed texts, minimal pairs in carrier phrases via the Diagnostic Rhyme Test or Modified Rhyme Test (DRT/MRT), or something known as Semantically Unpredictable Sentences (SUS), which are generated algorithmically to test whether quality breaks down in non-typical speech situations [5, pp. 314-315]. The human-transcribed output is typically corrected for orthographic errors before Word Error Rate (WER), a percentage of error between the output produced from the system and the transcribed speech, is calculated. The most state-of-the-art TTS systems currently available commercially are intelligible to the degree that they are indistinguishable from humans in dictation tasks [1].

Subjective Evaluation: Naturalness Naturalness is most frequently tested using a five-point Mean Opinion Score (MOS) test that has listeners evaluate voices and rate them with a score of how natural they sound. A similar intuition is to compare systems, either to each other or to a reference system. Given two systems, either systems that are different from each other or a system and a modification to that system, an AB test can present stimuli as randomly-ordered pairs of utterances A and B where the listener picks the better-sounding one. An ABX test is similar, and also has two different systems presenting randomly ordered stimuli, but instead of comparing A and B to each other, the listener compares them to system X and decides which of A or B is most similar to X [4].

2.2.1 A collaborative evaluation community: The Blizzard Challenge

The first Blizzard Challenge in 2005 was designed with the idea that data-driven (aka corpus-based) work in speech synthesis ought to be done at some level on shared data [6]. Because the time for voice creation using HMM-based synthesis is hours and not months or years as with unit selection, the creators of the Blizzard Challenge felt that having such a challenge in place would better offer a comparison of effectiveness of emerging research techniques [6]. The challenge offers a shared corpus from which participants build voices using their systems, and a unified subjective listening test hosted online [6]. While many early works on HMM-based synthesis use samples of the Japanese *ATR 503* database in similar preparations, comparing synthesis quality against an evaluation standard remains an outstanding issue despite community efforts to building experimental datasets (such as CMU-ARCTIC and others) to prove data-driven TTS synthesis [6].

3 Unit Selection–The Industry Standard

3.1 An overview of unit selection synthesis

Unit selection concatenative synthesis is waveform generation using a database of units of recorded speech which can be concatenated and joined. The basic intuition of unit selection synthesis is that natural, intelligible speech can be created by appropriate selection and concatenation, or stringing together, of units from a recorded speech database [7]. The database must be carefully recorded by a single-speaker and phonetically and phonemically normalized by planning a script that covers all the speech that will need to be stored. Signal processing is minimal in this method of waveform generation, and thus the style of the recording of the database is maintained in the output synthesis [7], [8], [9], [10]. What is put in can be taken out fairly reliably [11]. Signal processing degrades speech waveforms, and therefore, a secondary intuition is that less processing and fewer joins between units are best for producing high-quality unit selection synthesis [4], [8].

3.2 Unit selection voice quality

In spite of its simplicity, unit selection is effective. Unit selection evolved to be the dominant form of waveform generation and the industry standard for TTS synthesis, as early on, during the mid-1990s, it was able to produce more natural-sounding intelligible speech than model-based methods [1]. Unit selection can be perceptually indistinguishable from human speech, as based on subjective evaluation [9], and in limited domains where data can be designed to fit the domain, synthesis is excellent [11]. However, even one poorly selected unit can reveal the synthetic nature of the voice and significantly degrade perception of naturalness, and in extreme cases, intelligibility [1], [8]. The quality of synthesis is therefore directly dependent on the quality and consistency of the recorded database and the selection metrics used [8], [9].

3.3 Creating a unit selection database

Selecting appropriate units is the main challenge of unit selection. Unit selection means that there are many units of a given effective context (a perceptually-distinct phonetic and prosodic context) in a database, and the sequence that is the best fit globally (for the whole utterance) is chosen [7]. The more units, the more likely there is to actually be a good fit where a join can be made imperceptibly, but there will generally not be units that concatenate in the exact way that is given in the linguistic specification [4]. The drawback of more units is a larger footprint for the database and a more complex search task [7].

Context-sensitive units Typical systems use context-dependent diphones, which are created by cutting a phone-length unit from the environment of two adjacent phones. For example, given the phone sequence *axe*, diphones *ax* and *xe* would be created, each containing half of each phone. Diphones are made by cutting phones at the relatively spectrally-static middle of each phone to preserve spectral qualities of the intersection of the adjacent phones. Therefore the recordings must be phonetically-normalized to capture all contexts evenly [8]. A problem resulting from missing contexts is data sparsity, which leads to incomplete coverage of the needed database [11], [12]. Missing units can in some systems be interpolated using features from acoustically similar phones or units (rhyming, place of articulation, etc), or they can be dropped altogether. In either case, this coverage issue in the database affects quality by causing artifacts [4].

Unit size Units can take on many sizes. Though diphone units are the most common, different sized units have been used. Larger units like syllables have less joins but require more units to get good coverage, while smaller units (micro-units, sub-phone units) require more joins and a smaller footprint. The more joins there are, the more points of potential weakness are in the output from signal processing [13]. Different languages may have different best-size units for unit selection, which could affect the way the database may be best created to achieve the goal of TTS [13]

Reducing the unit selection voice footprint Attempts have been made to reduce unit selection databases by heavily preferencing units that are the most representative of a group of units while discarding others [14]. This method, known as clustering, is also able to pre-calculate the join cost of units by storing units in a classification and regression tree (CART) [14]. The CART uses a greedy algorithm to build a decision tree from the root based on best split of the data at every level [14]. Compression of the database size, or footprint, is desirable because it increases the number of applications that the synthesizer can be used in. The footprint of a database for a given voice must be small enough that it can fit into the system it is designed for, which means that judicious pruning using clustering methods is useful [14].

3.4 Selecting and joining appropriate units

Unit selection as it stands today was first truly formalized in [7] where the authors proposed using join (aka concatenation) costs and target costs. This means that a globally-best unit sequence can be chosen using Viterbi decoding of the database as a state transition network, with these costs as a way to connect every unit to every other unit via an overall cost function [7]. This means that where the linguistic specification matches the recordings taken, the overall cost to concatenate would be zero and the largest possible unit could be chosen. The intuition is that a lower overall cost will produce the best output by reducing signal processing [7], [10].

Figure 1 shows unit selection waveform generation from the linguistic specification (target in Figure) to the waveform output.

$$\mathit{Target\ Cost}: C^{(t)}(t_i, u_i) = \sum_{k=1}^{p} w_k^{(t)} C_k^{(t)}(t_i, u_i) \tag{2}$$

Target cost as in Equation (2)/Figure 1 measures distance between unit t_i in the linguistic specification and a unit u_i in the database using weighted linguistic features indexed by k [7].

$$\mathit{Join\ Cost}: C^{(j)}(u_{i-1}, u_i) = \sum_{m=1}^{p} w_m^{(j)} C_m^{(j)}(u_{i-1}, u_i) \tag{3}$$

Join cost as in Equation (3)/Figure 1 measures distance between a potential unit u_i and its immediately preceding unit u_{i-1} with weighted acoustic features indexed m. Note that this is not a static measurement; it is dependent on an unknown sequence of units, which is why Viterbi search is critical for finding globally optimal sequences [7].

$$\mathit{Overall\ Cost}: C(t_{1:n}, u_{1:n}) = \sum_{i=1}^{n} C^{(t)}(t_i, u_i) + \sum_{i=2}^{n} C^{(j)}(u_{i-1}, u_i) \tag{4}$$

Overall cost as in Equation (4)/Figure 1 is a sum of optimized target and join costs for a unit in a given context at a given time. It's simple to connect this to the $p(o|t)$, the probability of an observation o at time t and see how the Viterbi coding can work to select a single globally optimal sequence [7].

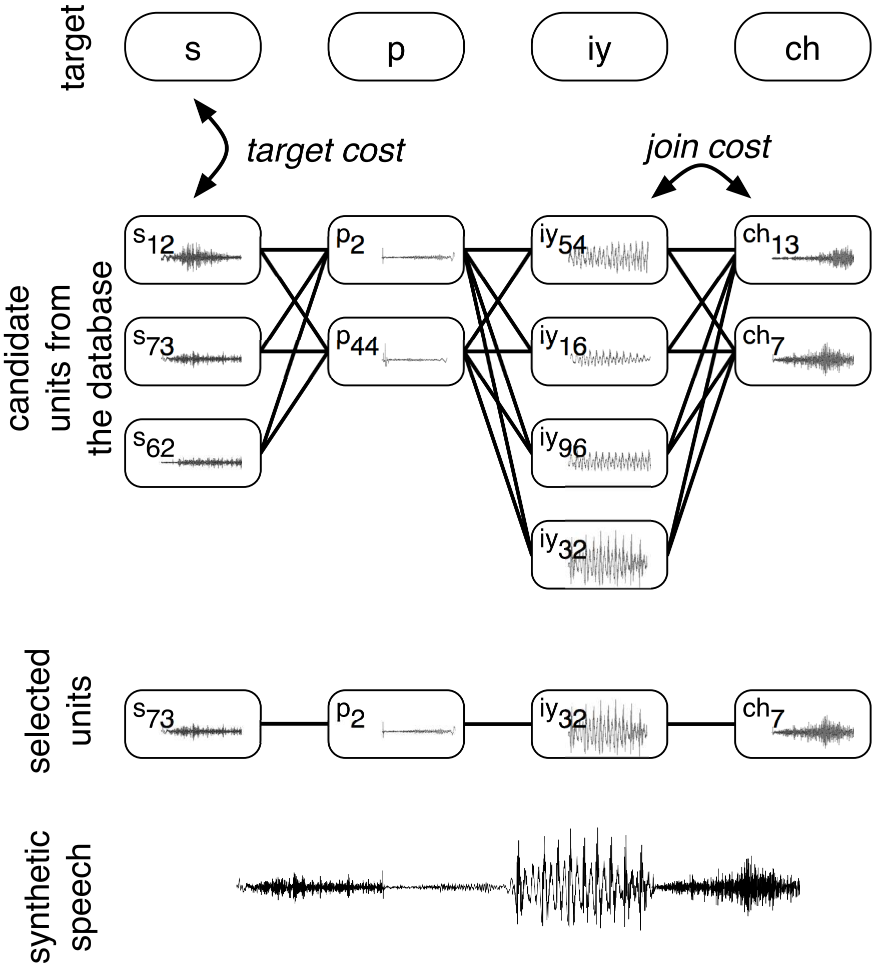

Figure 1: Simplified (non-context-dependent) visualization of unit selection based on Viterbi search of a state network using join and target costs. The “target” in this figure is the linguistic specification. Originally from [4].

Joining Though the idea of unit selection is to perform as little signal processing as is possible, the concatenated string of units selected will not fit together perfectly or match the linguistic specification in the desired way. Postproccessing, or smoothing joins is done to attempt to remove artifacts left by simply playing a string of units in order. Pitch Synchronous Overlap and Add (PSOLA) methods (e.g. Time Domain-PSOLA) can make minor modifications to fundamental frequency and durations but cannot do spectral smoothing. For smoothing, frequently some sort of parametric signal processing at the joins can interpolate an F0 and smooth formants effectively [4].

3.5 Advantages of unit selection synthesis

Currently, the overall high quality of unit selection is its biggest advantage as a method.

3.6 Limitations of unit selection synthesis

3.6.1 Rigidity/non-scalability

The major issue with unit selection is that it doesn't have very much flexibility to change nor scalable capacity to add styles or emotions without too-rapid expansion of the database.

Emotion and style Emotional speech and explicit style are examples of true limits for unit selection at present [1], [8]. A unit selection voice can only produce utterances in the style and/or emotion in which its database was recorded [8]. This means that from a database of whispering, it is impossible to get shouting, but moreover, that if there are not recordings of a given type in the database, it cannot be scaled to include them without recording and adding additional units, causing an explosion in the number of units a voice needs to have to cover the desired effect [8].

It is possible to synthesize a style with a given voice if the data is recorded in that same style with sufficient coverage of phonetic units [8], [9]. However, it can difficult even for professional voice talents to maintain a style or emotion consistently over the duration of recording; [8] uses the example of a voice talent who was unable to speak normally for several days after recording a small shouting database.

Approaches to creating a voice that is capable of handling multiple emotions or styles include tiering and blending. In tiering, styles are separated in a database and different domain-specific voices are accessed via a toggle-interface. In blending, all data is intermixed in one voice and specific units selected as needed [8], [9]. Blending is attractive because it could have a potentially smaller footprint, but tiered systems are more consistent in producing natural speech [9]

Prosodic manipulation In unit selection, the consistency of the recording conditions includes consistent prosody, meaning that unit selection actually includes an implicit model of prosody, and none is needed to supplement the prosodic information already included in a selected unit [9].

Rare Events Rare events in speech can never be covered adequately [12]. In order to have a unit section database that adequately models most rare events, the number of units has to greatly increase to grow the database proportionally, or the database will be heavily skewed to rare events [12]. This is the Achilles heel of unit selection because bad concatenation and bad joining are extremely perceptible, and nothing can be done that keeps a balanced database and also stores rare units.

3.6.2 High cost of labor-intensive voice creation

Making a single voice for unit selection can take days in a recording studio alone with a carefully prepared script and time for processing is much longer. The cost of a voice talent and the cost of a long and tedious process alone are very costly, but the expertise needed to make a commercial-grade unit selection system means that most companies can only afford to make a few full voices per language using unit selection [2], [15].

4 HMM-based Speech Synthesis–The State of the Art

4.1 An overview of HMM-based speech synthesis

HMM-based synthesis (aka statistical parametric speech synthesis) relies on the source-filter model of speech, and the intuition is that synthesis can be done using specialized HMMs that essentially take the average of all tokens for a given phonetic context from a corpus of data. The HMMs created are generative, meaning that after training (averaging across tokens), and after selection of a best-fit sequence of HMMs that match the linguistic specification, the models themselves generate spectral and fundamental frequency parameters, to which duration is added using a CART designed to model duration based on context [2]. With the spectral, fundamental frequency, and duration information, the basic TTS HMM is complete and excitation can be generated and passed through a filter based on the generated sequence of HMMs [1]. This is the basic model shown in Figure 2.

Early attempts in modeling used heuristic models of the vocal tract which had to be painstakingly hand-tuned for each voice and which ultimately still did not sound as natural as unit selection voices [16]. Modern modeling for TTS borrows frequently from ASR and statistical modeling, but less so from heuristics. HMM-based synthesis methods have been researched in their modern form for almost the same amount of time as unit selection methods, since the mid 1990s, but only in the last decade have they truly begun to achieve their potential for state-of-the-art synthesis [1].

4.2 HMM-based synthesis voice quality

Quality from HMM-based synthesis is improving and in some cases, it has been as natural and intelligible as unit selection [1], [17]. However, just because a voice is human-sounding does not mean that it will sound appropriately human [9]. HMM-based voices tend to have a very smoothed, damped quality [18].

4.3 Creating and improving a HMM-based synthesizer

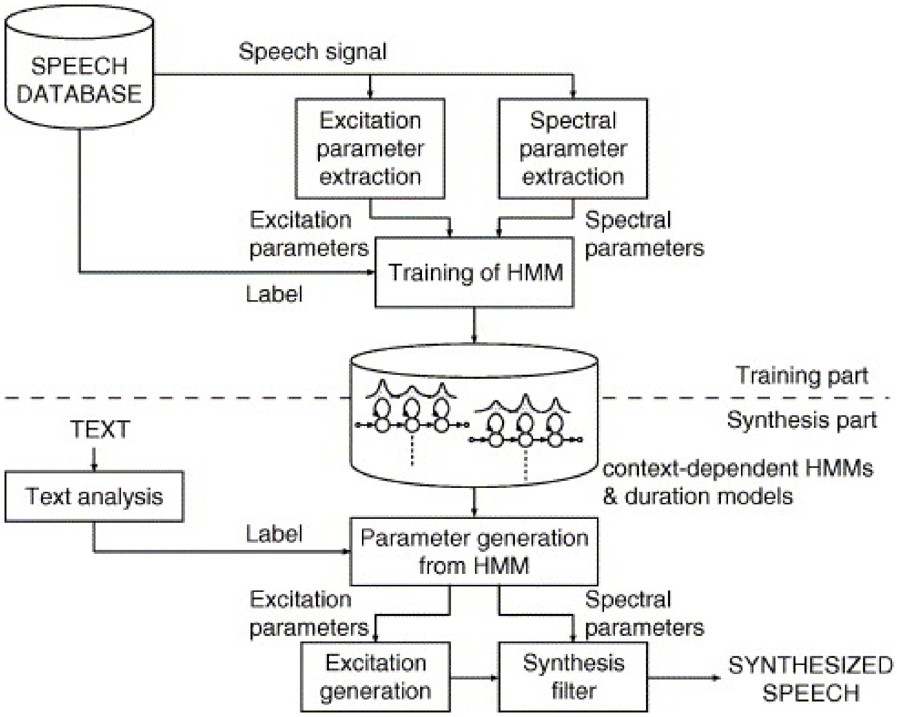

Figure 2: A simplification of a basic HMM-based speech synthesizer. Originally from [19].

HMM-based modeling as it is known today was first formalized around the same time as unit selection, in [20] and [21] which first showed the importance of modeling dynamic features (deltas, a measure of change between spectral characteristics over time) to improve speech quality. [20] and [21] spawned a steady stream of following papers which proposed using further dynamic features for synthesis (delta-deltas, a measure of change between deltas over time) [22], and crucially multi-streams [19]. HMM-based speech synthesis using multi-stream HMMs (HMMs where the observation vector consists of sub-vectors with static and dynamic features for spectrum, fundamental frequency, and state duration) was a pioneering step in 1999-2000 [2], and is the core way that HMM-based systems are modeled today [1], [19]. The ability to do "simultaneous modeling," as grounded in [19] and implemented algorithmically in [23], is a key concept that allowed for work that continued to improve on distinctiveness in parameter generation [1], [4], including parameters using HSMMs for duration modeling improvement [24], and trajectory modeling for continuous articulator movement modeling [25].

4.3.1 Training an HMM-based synthesizer

To train context-sensitive multi-stream HMMs from corpus data, the first task is performing forced alignment, a process that matches phones to their labels automatically. After labeling, groups of similar sounds are clustered and the Expectation Maximization (EM) algorithm uses iterative improvement to build a 3-5 state HMM with multi-streams. The excitation parameters (static: $logF0$; dynamic: deltas, delta-deltas) and the spectral parameters (static: Mel-frequency cepstral coefficients (MFCCs) or similar; dynamic: deltas, delta-deltas) are modeled in separate statistically independent streams within the same HMM. The training process is much like in automatic speech recognition (ASR), but with more information modeled in the HMM framework. The simple reason for modeling more information is that speech synthesis needs to generate a complete representation of a speech waveform, while ASR has the goal of reducing as much information as possible for speed and performance [19]. Training in Figure 2 takes place in the top half of the figure. Looking at the diagram, it is easy to see that with so much signal processing, using good parameters that can capture and extract speech parameters that can be resynthesized well is crucial to achieving best quality.

4.3.2 Modeling improvements

Modeling improvements have one central goal in HMM synthesis, which is to increase the distinctiveness of individual models or phones. Below are some but certainly not all influential model improvements from the last decade of HMM-based synthesis.

Duration modeling with HSMMs Duration modeling in standard non-speech HMMs is exponential, based on remultiplication of the same self-transition probability for a given state. This is not an adequate model of duration for speech synthesis, as some types of sounds (i.e. vowels) are consistently longer in duration than others (i.e. stops), though duration is not a discriminative feature in its own right because many phones have similar durations [4].

The current state-of-the-art in duration modeling for HMM-based synthesis is the Hidden Semi-Markov Model (HSMM), a method that uses the context of a previous model for explicit duration modeling as a (typically) log-Gaussian distribution [24]. In experiments [17], [24], HSMMs significantly improve the naturalness of HMM-based synthesis.

Trajectory Modeling Using static spectral features alone in making context generalizations does not account for the continuous movement of the articulators in natural speech [4]. Trajectory HMMs, as introduced in [25] attempt to account for this continuous movement as context by defining ”explicit relationships” between static (spectral) and dynamic (delta and delta-deltas) features [25]. These relationships mean that the degree of change, as indicated by dynamic features, is included in the context of a model. This improvement aids in discrimination between models and was shown to provide significantly more natural speech (based on subjective evaluation) as well as better performance in automatic speech recognition, with the advantage of being able to be included in existing research frameworks; no substantially new algorithms or methods need to be used to create trajectory HMMs [4], [25].

4.4 Generating speech from a HMM-based synthesizer

Referring again to Figure 2, the lower half of the diagram, labeled as the "synthesis part," operates in a way similar to training. Recall that in training, labeled speech data goes in and is parameterized and modeled. In synthesis, models are selected which optimally fit the linguistic specification, and they are resynthesized by generating excitation and passing through a filter called the MLSA, or Mel-cepstral resynthesis.

4.4.1 Excitation

Basic excitation, a pulse train and white noise, imparts buzziness and is inadequate for high-quality HMM-synthesis [17], [19]. There are a variety of excitation methods, but out of the three described here, STRAIGHT, an older but still widely-used mixed-band excitation method, and MELP, or mixed excitation linear prediction, are the two most commonly used [1]. Maximum Likelihood (ML) excitation is one of the newest excitation methods [1]. ML excitation uses residuals to derive high quality signal from a natural speech signal.

4.5 Advantages of HMM-based synthesis

4.5.1 Flexibility

Flexibility is one of the greatest advantages of HMM-based synthesis. Because the parameters can be manipulated, new voices can be made using statistical transformations relatively easily and in a fraction of the time of unit selection voices [1], [2]. New voices can be made using speaker adaptation, speaker interpolation, and eigenvoice techniques, among others. As a sample only, the arguably most popular speaker adaptation technique is described below.

Speaker adaptation Speaker adaptation is a general group of techniques used to take the characteristics of a target speaker's voice and use them to adapt an existing HMM-system to better fit their voice. A speaker-adaptive HMM-based synthesis system is given in Figure 3, as taken from [26]. In the Figure, the center adaptation portion shows where target-speaker characteristics come in, but not precisely how. Speaker adaptation can be done in a number of ways, but probably the most common way (designed for ASR originally) is to use maximum likelihood linear regression (MLLR) [4]. MLLR allows for supervised adaptation even from a small amount (a few minutes) of target speaker data by extracting relevant speaker characteristics, such as vocal tract shape and fundamental frequencies and applying a linear adaptive transform to selected correlating parameters in the trained HMMs [27]. Looking back at Figure 3, what precisely is happening is

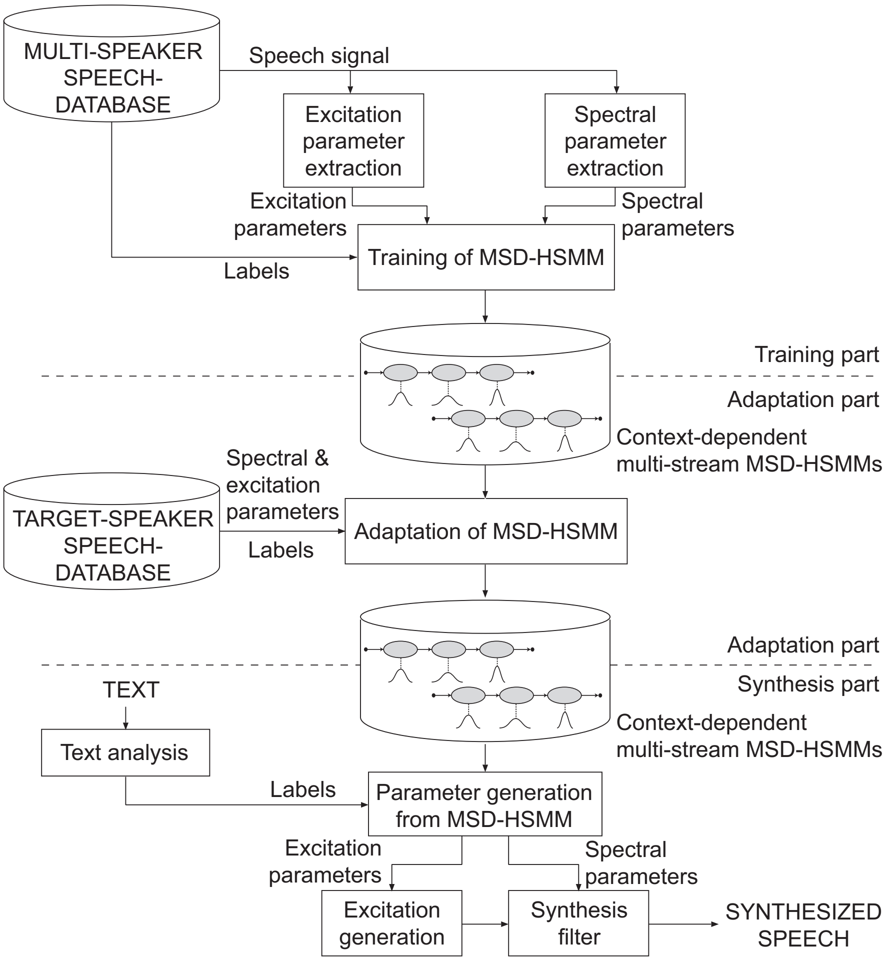

Figure 3: Adaptive HMM-based speech synthesizer. Originally from [26]

that some amount of data from the target speaker is being injected into the already-trained system, replacing the data where the target speaker's personal characteristics are captured.

4.5.2 Control

HMM-based synthesis allows for independent control over the spectral and fundamental frequency modeling parameters, meaning aspects of the voice can be filtered out and changed individually [17]. Speaker adaptation and interpolation exploit this fact to use MLLRs to change the voice in a linear way for simple adaptation or interpolation [27].

4.5.3 Footprint

The footprints of HMM-based synthesizers can be many times smaller than the footprints of unit selection systems built on the same data, and this is because the HMM-based models do not need to store any actual speech, just a model of speech. For embedded systems and mobile platforms with limited available space, HMM-based synthesis could be the best tool for TTS in applications.

4.5.4 Robustness

Unbalanced data and Noisy Data Phonemically unbalanced recording data is non-standardized data that doesn't necessarily cover all contexts adequately. Noisy recording data is data that contains an unknown amount of ambient noise and reverberation. These effects can cause problems for creating a consistent voice using the parameters from that data in HMM-based synthesis if not accounted for. Some very recent work has been done that showcases the flexibility of HMM-based synthesis, in that reasonable voices were created using such data as televised speech from an American politician [28], ASR corpora [15], and audiobooks [29]. These experiments used particular speaker-adaptive techniques to isolate parameters from the inconsistent data and transform an HMM-based system to create a voice or many voices [28].

Robust training is an area of HMM-based synthesis that has received attention since [28], as the ability to model speech using speech with degradation of the signal has the potential to open up a consumer market for TTS systems on a variety of platforms where clean and balanced data capture would be impossible.

4.6 Limitations of HMM-based synthesis

The main limitation of HMM-based synthesis is naturalness [1]. Quality in such synthesis suffers from heavy signal processing, and without appropriate models, synthesis is highly degraded.

5 Unit Selection/HMM-based Hybrids–The best of both worlds

It is clear that unit selection and HMM-based methods both offer advantages and disadvantages. The intuition of a new and progressively more popular area of research in hybrid data-driven systems is that the right system could retain all the advantages and minimize the disadvantages of both methods [1].

A variety of types of systems have been proposed and tested, although evaluation is often not comparable in some experiments. Though this review does not give a full coverage of hybrid systems, some interesting ideas include interpolating and generating units from HMMs when appropriate units are not available in unit selection [30], and actually performing selection on HMM states as sub-phone units [31].

In [1], Zen et. al. express the idea that hybrids are perhaps the way to overcome some of the problems in unit selection and HMM-based synthesis and that the number of approaches tried so far is just a beginning to the number of approaches to try. As HMM-based methods evolve, it will become apparent what the role of hybrids is in TTS.

6 Conclusions

This review has considered the recent co-evolution of the two predominant methods for contemporary data-driven waveform generation, unit selection synthesis and HMM-based synthesis. The review highlights advantages and disadvantages to each considering potential in research and industry for current and future high-quality synthesis. The review has also cited research trends and advances, such as the proliferation of hybrid methods. Based on evidence from the investigation in this review the following is a comparison of unit selection and HMM-based synthesis.

Unit selection synthesis is the current industry standard, and the goal of TTS, a minimum error equivalent to real speech has been achieved in limited domain systems using unit selection. On its best day, an HMM-based synthesizer would be hard-pressed to hold up in naturalness to an expensive commercial unit selection voice. Still too, HMM-synthesis suffers from a lack of variability that comes from natural speech, and in general, though HMM-based synthesis has found success, it is still not reliably natural or prosodically refined.

However, HMM-based synthesis shows the obvious greatest potential to continue to dominate research and perhaps even industry in a few years because of its unparalleled flexibility, robustness, and low cost in a (relatively) small footprint. Research in unit selection will continue to look for a way to best select units in a framework where bad joins are no longer the restricting factor. Because of rare events in speech, it is not possible to create a reasonable sized unit selection voice that covers all the data adequately to avoid poor synthesis entirely. Beyond rare events though, unit selection cannot keep up with the influx of data available to model emotional speech.

HMM-based synthesis offers the flexibility and control that unit selection synthesis cannot. Though HMM-synthesis might face naturalness challenges now, there is a huge potential to mold the synthesizer in any way required using the fine-grained parametric control inherent to statistical modeling. As Zen et. al. assert in [1], quality in HMM-based synthesis is a parameter issue–though since the Blizzard Challenge in 2005, some HMM-based voices including [17] have been able to stand on their own against unit-selection voices and match or exceed quality as measured by dictation tasks and the 5-point MOS.

The evidence for the potential of HMM-synthesis is clear; the future of high-quality synthesis and the overcoming of naturalness issues, for a variety of reasons, seems to lie in improving modeling methods.

7 Future Work

7.1 Future work in unit selection

In [1], the authors purport that work in unit selection will continue to focus on better cost evaluations, distance measures, feature weighting methods, and algorithms; essentially, on ensuring the the best units are selected. With the emergence of embedded TTS systems in the commercial mobile market, it is likely that work in decreasing the footprints of unit selection systems through data compression and through database pruning will also continue [14].

7.2 Future work in HMM-based synthesis

Natural speech has variability, and one area where HMM-based methods still struggle is in generating believable variability [4]. It raises the question of whether maximum likelihood generation from HMMs is adequate or if it smooths the output speech too much [18]. Generally, better articulatory models can also help in this area, by perhaps illuminating some unknown causes of variability that do fit in the HMM framework. Prosodic and style modeling and experiments with TTS, such as [29], are likely to be a helpful source of variability. On a more evaluative level, work attempting to figure out why some systems are preferred over others might open up ways to improve naturalness.

HMM-based synthesis is already showing great promise for embedded system applications, with the smallest individual voice footprints at around 100kB [1]. Additionally, the greater flexibility of speaker adaptation means that there is untapped potential for creating user-specific embedded voices, which could have a wide range of uses in applications from improved donor-banked assistive speech system voices [32] to widening the selection of voices for every application from mobile map-reading to video game characters.

Work has been done to largely cement a model of the spectral parameters, but work remains in excitation source modeling. Current methods for the most part impart "buzziness" or other audible artifacts of processing to the voice, hindering naturalness. A new and exciting area of HMM-based synthesis research looks at robustness and includes making use of data which is unbalanced and/or noisy to make voices.

7.3 Future work towards data-driven hybrid synthesis

The future of the hybrid system is unknown but as evidenced by the number of different hybrid approaches, and the ambition of hybrids to capitalize on the best of both worlds of data-driven synthesis, the field may be a fertile area [1].

References

[1] Zen H., Tokuda K., and Black A. 2009. Statistical Parametric Speech Synthesis. In *Speech Communication, 51(11), pp 1039-1064, November 2009*

[2] Black A. W. 2006. CLUSTERGEN: A Statistical Parametric Synthesizer using Trajectory Modeling. In *Proc. Eurospeech, pp, 1762-1765.*

[3] Fordyce C. S. and Ostendorf M. 1998. Prosody prediction for speech synthesis using transformational rule-based learning. In *Proc. International Conference on Spoken Language Processing (ICSLP) 1998*

[4] Gold B., Morgan N., and Ellis D. *Speech and Audio Signal Processing, Ch. 29: Speech Synthesis.* Wiley, 2nd Edition, 2011.

[5] Jurafsky D. and Martin J. H. *Speech and Language Processing.* Prentice Hall, 2nd Edition, 2009.

[6] Black A. W. and Tokuda K. 2005. The Blizzard Challenge – 2005: Evaluating Corpus-Based Speech Synthesis on Common Datasets. In *Proc. Interspeech 2005. Lisboa*

[7] Hunt A. and Black A. W. 1996. Unit selection in a concatenative speech synthesis system using a large speech database. In *Proc. ICASSP. pp. 373-376*

[8] Black A. W. 2003. Unit Selection and Emotional Speech. In *Proc. EUROSPEECH. pp, 1649-1652*

[9] Black A. W. 2002. Perfect synthesis for all of the people all of the time. In *Proc. IEEE Speech Synthesis Workshop.*

[10] Black A. W. and Campbell N. 1995. Optimising selection of units from speech databases for concatenative synthesis. In *EUROSPEECH '95, pp. 581-584, Madrid, Spain.*

[11] Black A. W. and Lenzo K. 2000. Limited Domain Synthesis. In *Proc. ICSLP. pp.411-414.*

[12] Möbius B. 2003 Rare events and closed domains: Two delicate concepts in speech synthesis. *International Journal of Speech Technology 6 (1), 57-71.*

[13] Kishore S. P. and Black A. W. 2003. Unit Size in Unit Selection Speech Synthesis. In *Proc. Interspeech, pp. 1317-1320*

[14] Black A. W. and Taylor P. 1997. Automatically clustering similar units for unit selection in speech synthesis. In *Proc. Eurospeech. pp. 601-604.*

[15] Yamagishi J., Usabaev B., King S., Watts O., Dines J., Tian J., Guan Y., Hu R., Oura K., Wu Y., Tokuda K., Karhila R., and Kurimo M. 2010. 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, 18(5):984-1004, July 2010.*

[16] Klatt D. 1987. Review of text-to-speech conversion for English. *Journal of the Acoustic Society of America., vol. 82, pp.737 -793.*

[17] Zen H. and Toda T. 2005. An Overview of Nitech HMM-based Speech Synthesis System for Blizzard Challenge 2005. In*Proc. Interspeech 2005.*

[18] Toda T. and Tokuda K. 2005. Speech Parameter Generation Algorithm Considering Global Variance for HMM-Based Speech Synthesis. *IEICE Trans. Inf. Syst. E90-D (5), 816-824.*

[19] Yoshimura T., Tokuda K., Masuko T., Kobayashi T., and Kitamura T. 1999. Simultaneous modeling of spectrum, pitch and duration in HMM-based synthesis. In *Proc. Eurospeech. pp. 2347-2350.*

[20] Tokuda K, Kobayashi T., and Imai S. Speech parameter generation from HMM using dynamic features. In *Proc. International Conference on Acoustics, Speech, and Signal Processing 1995* (1995).

[21] Tokuda K., Masuko T., Yamada T., Kobayashi T., and Imai S. 1995. An algorithm for speech parameter generation from continuous mixture HMMs with dynamic features. In *Proc. of EUROSPEECH, pp. 757-760.*

[22] Masuko T., Tokuda K., Kobayashi T., and Imai S. 1996. Speech synthesis using HMMs with dynamic features. In *Proc. of ICASSP, pp. 389-392.*

[23] Tokuda K., Yoshimura T., Masuko T., Kobayashi T., and Kitamura T. 2000. Speech parameter generation algorithms for HMM-based speech synthesis. In *Proc. ICASSP. pp 1315-1318.*

[24] Zen H., Tokuda K., Masuko T., Kobayashi T., and Kitamura T. 2004. Hidden Semi-Markov Model Based Speech Synthesis. In *Proc. ICSLP. 1397-1400*

[25] Zen H., Tokuda K., and Kitamura T. 2007. Reformulating the HMM as a trajectory model by imposing explicit relationships between static and dynamic feature vector sequences. *Computer Speech & Language. Vol. 21, 1. pp. 153-173.*

[26] Yamagishi J., Kobayashi T., Nakano Y., Ogata K., and Isogai J. 2009. Analysis of Speaker Adaptation Algorithms for HMM-Based Speech Synthesis and a Constrained SAMPLR Adaptation Algorithm. In *IEEE Trans. Speech Audio Lang. Process. 17 pp. 66-83*

[27] Leggetter C. J. and Woodland P. C. 1995. Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models. *Computer Speech & Language. Vol. 9. pp. 171-185*

[28] Yamagishi J., Ling Z., and King S. 2008. Robustness of HMM-based speech synthesis. In *Proc. Interspeech. pp. 581-584*

[29] Chen L., Gales M. J. F., Wan V., Latorre J., and Akamine M. 2012. Exploring Rich Expressive Information from Audiobook Data Using Cluster Adaptive Training. In *Proc. Interspeech*

[30] Pollet V. and Breen A. 2008. Synthesis by Generation and Concatenation of Multiform Segments. In *Proc. Interspeech. 1825-1828*

[31] Taylor P. Unifying Unit Selection and Hidden Markov Model Speech Synthesis. In *Proc. International Conference on Spoken Language Processing (ICSLP) 2006.* 2006.

[32] Yamagishi J., Veaux C., King S., and Renals S. 2012. Speech synthesis technologies for individuals with vocal disabilities: Voice banking and reconstruction. *Acoustical Science and Technology 33.1 (2012): 1-5.*