Adaptation and Frontend Features to Improve Naturalness in Found-Data Synthesis

Erica Cooper, Julia Hirschberg

Columbia University, USA

ecooper@cs.columbia.edu, julia@cs.columbia.edu

Abstract

We compare two approaches for training statistical parametric voices that make use of acoustic and prosodic features at the utterance level with the aim of improving naturalness of the resultant voices – subset adaptation, and adding new acoustic and prosodic features at the frontend. We have found that the approach of labeling high, middle, or low values for a given feature at the frontend and then choosing which setting to use at synthesis time can produce voices rated as significantly more natural than a baseline voice that uses only the standard contextual frontend features, for both HMM-based and neural network-based synthesis.

Index Terms: Speech synthesis, parametric synthesis, prosody, found data, crowdsourcing.

1. Introduction

Recent advances in speech technology have led to a proliferation of speech-enabled applications. From virtual assistants such as Apple's Siri and Amazon's Echo to Panasonic's translating airport megaphone and Samsung's talking refrigerator, it may appear that speech technology is everywhere. However, this is only truly the case for languages which have received the corporate or government resources and research attention required to collect and annotate the large amounts of data and linguistic resources needed to build precise, domain-appropriate speech models. Text-to-speech (TTS) synthesis is a key component of interactive, speech-based systems, and typically, building a high-quality voice requires collecting many hours of speech from a single professional speaker in an anechoic chamber with a high-quality microphone. There are over 6,000 languages in the world, and most do not enjoy the speech research attention historically paid to such languages as English, Spanish, Mandarin, and Japanese; speakers of many other languages therefore do not benefit equally from these technological advances. They are thus deprived of technologies they can use to communicate and search for information in their own language by voice — a major accessibility issue for those who lack the ability to read. Furthermore, as we move towards becoming a more global world, access to language technologies such as speech translation becomes important not only for travelers but for medical professionals, emergency response staff, corporations, journalists, the military and law enforcement.

While it takes a great deal of time and resources to collect a traditional text-to-speech corpus for a given language, we may instead be able to make use of various sources of "found" data on the web or collected for purposes other than TTS. In particular, sources such as radio broadcast news, audiobooks, podcasts, and data collected to train automatic speech recognition (ASR) engines are available in many languages. While this type of data is quite different from data one would collect for a standard TTS corpus, it may nevertheless contain a substantial amount of speech from each speaker, the speakers may be professionals or at least demonstrate consistency in their speech, and the recording conditions may be fairly high-quality. The major difference in data collected specifically for TTS corpora is that TTS speakers are typically instructed to speak as consistently as possible, without varying their voice quality, pitch range, speaking style, volume, or tempo significantly [1], whereas even broadcast news anchors will deliberately introduce some variation in their speaking style to produce more engaging speech, even when they are otherwise speaking in a predominantly neutral style. The innovation in TTS research of statistical parametric synthesis [2] has enabled the use of such variable sources of data for building intelligible and natural-sounding TTS voices.

Others have used found data for creating TTS voices, exploring various approaches to select or otherwise manipulate the data to be more similar to what one might find in a traditional TTS corpus. Political speeches [3], radio broadcast news [4], and audiobooks [5, 6, 7] have all been popular sources. We are specifically interested in radio broadcast news not only because it contains large amounts of speech from each anchor and is often professionally recorded, but because it is available in many languages. In our previous work [8, 9], we explored data selection and outlier removal at the utterance level to produce voices that are rated as more natural, even though they were trained on a smaller amount of data than a baseline trained on all of the data. We selected our subsets based on a number of different acoustic and prosodic features, finding that removing outliers for hyper-articulation and combining filters for hypo-articulation and low mean f0 produced voices rated as significantly more natural. Since the data we used in this research, the Boston University Radio News Corpus (BURNC) [10], is high-quality, we have also examined whether we could benefit from using all of the data while also benefiting from these informative features by using a subset-adaptive training approach. In this approach, we treated all of the utterances in a feature-selected subset as one regression class and all other utterances as another, adaptively training the voice, and then adapting the average voice model towards the subset class. While this approach produced slight improvements for naturalness over the baseline for some features, none were statistically significant.

In the current work reported here, we expand upon our subset adaptation approach, as well as another approach that makes use of all the data from a corpus: rather than treating ranges of these feature values as regression classes and adaptively training, we add in the ranges for these feature values at the frontend. Each utterance is labeled as "high," "middle," or "low" for each feature, such that the data is divided into thirds. Then, we synthesize our test utterances using each of those three settings. There is a precedent for this approach in [11], where the authors used an approach called "style mixed modeling" to train a voice that could speak in different styles. This entailed using data from one speaker speaking in different styles, labeling the styles at the frontend along with the standard set of linguistic features, and then choosing which style to synthesize with at output, by including the style along with the other standard contextual features. In our work, rather than having predefined speaking styles, we are using measurable acoustic and prosodic features and treating ranges of each of these as their own "manner of speaking," aiming for speech that is close to more typical TTS data in order to produce the most natural synthesis.

One limitation of our prior work [9] was the unequal and split nature of the regression classes. When adapting to a "middle range" subset, the "not-in-subset" class is comprised of two disparate parts of the data — low- and high-valued utterances for that feature. Combining these may have harmed the model's consistency. Thus, in this work, we explore the use of three classes, "high," "middle," and "low," each consisting of a third of the data, rather than using fixed hour-long subsets. We hypothesize that this approach will produce more consistent models with better ability to adapt towards the desired speaking characteristic, in addition to enabling a more direct comparison to the "style mixed modeling" frontend-labeling approach using identical partitions of the data.

2. Tools and Corpus

We initially trained our TTS voices using the Hidden Markov Model Based Speech Synthesis System (HTS) [12], version 2.3, using the hts_engine vocoder. However, with recent advances in neural network based speech synthesis, we also wished to learn which results generalize across acoustic model types, so we repeated our adaptation and frontend-labeling experiments using the Merlin [13] toolkit for neural network based voice training, with the WORLD [14] vocoder. For text processing, we used the default U.S. English frontend for Festival [15]. Although we ultimately aim to build TTS voices for LRLs, we initially use US English data for our pilot work to facilitate evaluation and experimental iteration.

Our training data is again the Boston University Radio News Corpus (BURNC) [10] which consists of professionally read radio news from four male and three female FM radio news announcers associated with the public radio station WBUR. The main corpus includes news recorded in the station's studio during broadcasts over a two-year period. In addition, the same announcers were recorded in a laboratory in both non-radio and radio speaking styles. We used the broadcast portion of the corpus with the orthographic transcriptions for our experiments. We trained voices using only the 4 hours and 22 minutes of female data in order to produce more consistent models. We segmented the data into utterances, defined as sentences.

We evaluated all of our voices for naturalness using Amazon Mechanical Turk (MTurk), a popular crowdsourcing platform. To restrict our task to native speakers of English, we required workers to complete a qualification test first, in which they had to identify the languages they have spoken since birth from a list of options. We only allowed workers who selected English and no more than two other languages to participate, in order to exclude those who might select, e.g., all of the languages in an attempt to game the system. We also restricted our tasks' visibility to workers within the United States. The task consisted of a pairwise comparison between the baseline voice and a test voice. Each task thus contained only two audio files, the same sentence spoken by the baseline voice and by one of our test voices. Workers could rate as many or as few pairs of utterances as they wished. Half of the sentences were presented in A/B order and the other half in B/A order, to avoid possible order effects. We ensured that raters played both audio files entirely. Raters were given a forced choice, i.e. there was no "no preference" option. We chose 12 lexically neutral sentences of varying length from the fable "Jack and the Beanstalk" and synthesized each of them with our voices. Each task was completed by 5 workers, for a total of 60 comparison ratings for each voice.

3. Acoustic and Prosodic Features

We explored features related to manner of speaking, namely mean and standard deviation of f0 and energy, identified automatically using Praat [16]; speaking rate in syllables per second; level of articulation, defined as mean energy divided by speaking rate; and duration of the utterance. For each of these features, we sorted our training utterances by feature value and then divided the data into thirds, labeling each utterance as having a high, medium, or low value for that feature as appropriate. For the adaptation approach, we treated each third of the data in turn as an adaptation set and used it to adapt an average voice model (AVM). For the frontend labeling approach, we introduced a new contextual feature, added on to the standard set of contextual features for English. This new frontend feature took on the value of high, middle, or low as appropriate for each utterance. We also added our new contextual features to the test output labels, creating high, medium, and low-setting label files, in order to compare synthesis output at each setting. Our baseline was a voice speaker-independently trained on all of the female data with only the standard contextual features.

4. HMM-based Synthesis Experiments

4.1. Adapted Voices

The BURNC corpus contains relatively high-quality speech, so we explore approaches that use all of the data, while also making use of our informative acoustic and prosodic features. One way to do this is to treat each subset of high, middle, and low-valued utterances for each feature as a separate regression class. We adaptively trained one voice per feature, and then synthesized test utterances adapted to each of the three classes. To accomplish this, we used the HTS speaker-adaptive training recipe, but instead of labeling different speakers, we labeled high, middle, or low values for the given feature. Results are shown in Table 1, with best settings for each feature in bold.

Table 1: *Percent preference for HTS voices trained adaptively using high, middle, and low partitions for each feature.*

Feature	hi	med	lo
Mean f0	40.0	53.3	**56.7**
Std. dev f0	33.3	38.3	**43.3**
Mean energy	41.7	**60.0**	58.3
Std. dev energy	**43.3**	41.7	40.0
Speaking rate	**46.7**	**46.7**	35.0
Articulation	38.3	30.0	**40.0**
Duration	**40.0**	31.7	36.7

While low mean f0 and middle mean energy adapted voices were rated as better than the baseline, neither of these preferences turned out to be statistically significant.

4.2. Contextual Feature Labeled Voices

Another way to make use of all of the data while also making use of informative acoustic and prosodic features is to label each utterance as having a high, middle, or low value for a given feature at the *frontend*, as part of the set of contextual features. One major benefit of this approach is that, in the construction of the decision trees for Hidden Markov Model based synthesis, if there are any contextual features that are not actually informative in splitting the data, they simply will not be used. Therefore, we are able to add arbitrarily many new contextual features, which, if they do not contribute to better modeling of the data, simply will not appear in the decision trees.

The standard set of contextual features is obtained using Festival, and includes phoneme-level information such as the previous two, current, and next two phonemes; the position of the current phoneme in the syllable; position of the current syllable in the word; whether the syllable is stressed or not; position of the current word in the phrase; and similar features providing a linguistic representation. Using the same partitions of the data into thirds, we added one new contextual feature to our fullcontext labels, indicating whether the utterance has a high, middle, or low value for one particular feature; we also added relevant questions to the HTS *questions file*. The questions file for HTS voice training contains a variety of yes/no questions that are used in the construction of the acoustic model decision tree, each followed by patterns for which a match in the full-context label would indicate a "yes". We added three new questions that ask whether the new feature value is high, medium, or low. We then trained one voice for each feature on all of the data labeled as described, and then synthesized test utterances with each of the three settings. Results are shown in Table 2, with the best setting out of high, medium, or low in bold, and statistically-significant preferences underlined.

Table 2: *Percent preference for HTS voices trained with labels for high, medium, or low values for acoustic and prosodic features and then synthesized at each of the three settings.*

Feature	hi	med	lo
Mean f0	55.0	**60.0**	51.7
Std. dev f0	60.0	55.0	**63.3**
Mean energy	48.3	**56.7**	45.0
Std. dev energy	**51.7**	50.0	**51.7**
Speaking rate	**50.0**	46.7	45.0
Articulation	**56.7**	**56.7**	**56.7**
Duration	**63.3**	50.0	56.7

Synthesizing with the low setting for standard deviation of f0 and with the high setting for duration both produced speech that was significantly preferred over the baseline. The success of the low standard deviation of f0 setting makes sense because professional speakers for a TTS corpus are typically instructed to speak with as little variation as possible [1]. For the "high duration" synthesis, we are not necessarily synthesizing long utterances, but rather choosing to synthesize *in the style of* the longer utterances in the training data. This may have resulted in better naturalness ratings because longer training utterances provide more speech in a natural context.

Next, we wanted to see whether combining features could produce even more improvement. Rather than trying all combinations and all settings of the features, we accumulated features one by one in the order that they gave improvement, and only synthesized using the best setting for each feature.

For some features it was not clear which setting was "best" – in particular articulation and standard deviation of energy. So we posted tiebreaker HITs on MTurk. The tie was not resolved for articulation, so we picked the low setting, corresponding with our prior findings [8, 9] that training on hypo-articulated utterances tends to produce better voices. For standard deviation of energy, the low setting was slightly preferred.

Synthesizing with only the best setting for each feature, our features gave improvements over the baseline in the following order, from most to least: duration (hi), standard deviation of f0 (lo), mean f0 (med), articulation (lo), mean energy (med), standard deviation of energy(lo), and speaking rate (hi). We thus trained six new voices: the first, with both duration and standard deviation of f0 labeled in the contextual features and with the "hi" and "lo" settings for those features, respectively, chosen at synthesis; the next voice, with those same features plus mean f0, set to the "med" setting at synthesis; and so on. Preferences over the baseline are shown in Table 3; note that each line of the table represents features from the preceding line *plus* the new feature added on the current line.

Table 3: *Percent preference for HTS voices trained with labels for multiple features*

Features	Preference
Duration (hi) + Std. dev. f0 (lo)	46.7
+ Mean f0 (med)	53.3
+ Articulation (lo)	56.7
+ Mean energy (med)	58.3
+ Std. dev. energy (lo)	65.0
+ Speaking rate (hi)	60.0

Surprisingly, the best two features, which on their own resulted in better voices (Table 2), produced a worse voice in combination. We see improvements as we add each feature, with the exception of adding speaking rate, which results in a slight drop in naturalness ratings. These features appear to be interacting in unexpected ways, which we must examine further.

5. Neural Network Synthesis Experiments

Neural network based synthesis has recently produced very high-quality voices, and addresses some of the naturalness issues common to HMM-based voices. [17] found that the across-class averaging resulting from decision tree based context clustering is a major detractor of naturalness in HMM voice quality, and [18] found that replacing the decision trees with DNNs and the production of frame-level rather than state-level predictions substantially improved naturalness as well. Furthermore, [19] found that an HMM system trained on 100 hours of data was comparable in f0 correlation (an objective measure of naturalness) to a DNN system using only 10 hours. While these results were for voices trained on single-speaker data collected specifically for TTS, we would also like to explore modeling approaches that can produce higher-quality voices with less data. Thus, we have begun neural network-based voice training in addition to HTS in order to determine experimentally whether these advances generalize to the type of data we are using.

We repeated our experiments using the Merlin toolkit for neural network based synthesis [13]. For the baseline and frontend-feature experiments, we used the basic "build your own voice" recipe from Merlin, using WORLD for feature extraction and vocoding. These models consist of 6 TANH layers each of size 1024, with a linear activation function at the output layer, and a batch size of 64 for the duration model and 256 for the acoustic model. Learning rate was fixed at 0.002, momentum was 0.3, and number of training epochs was 25.

First, we trained a baseline voice using this recipe with all of the female BURNC data using the standard fullcontext labels extracted by Festival. When we compared this to the HTS baseline using the same audio and labels, the preference for the Merlin voice was **90.0%**. It is therefore apparent that not only does neural network based synthesis produce more natural voices when trained on standard TTS data, but on mixed, found data from radio broadcast news as well.

5.1. Adapted Voices

For the adaptation experiments, we trained an AVM on all of the female data and then adapted to each subset using the Merlin speaker adaptation recipe, which implements two different types of adaptation (described in [20]): back-propagating the adaptation data through the model to re-tune all the weights ('fine-tune'); and "Learn Hidden Unit Contributions" (LHUC), which recombines hidden units based on the adaptation data [21]. We tried both methods, and we found that the best voices were produced using the 'fine-tune' adaptation method; full results for fine-tune adapted voices are shown in Table 4.

Table 4: *Percent preference for Merlin AVM adapted to subsets of the data selected based on high, middle, or low values for various acoustic and prosodic features.*

Feature	hi	med	lo
Mean f0	43.3	**45.0**	36.7
Std. dev f0	48.3	**60.0**	50.0
Mean energy	**53.3**	45.0	36.7
Std. dev energy	36.7	**43.3**	36.7
Speaking rate	**45.0**	**45.0**	41.7
Articulation	**50.0**	45.0	45.0
Duration	41.7	45.0	**60.0**

Adapting to short duration utterances and adapting to middle standard deviation of f0 were both preferred over the baseline by 60%, which was not statistically significant.

5.2. Contextual Feature Labeled Voices

We repeated our experiments from Section 4.3, adding one new feature at the frontend that takes on a value of high, medium, or low depending on the utterance's value for the given acoustic or prosodic feature we are measuring, and then synthesizing output utterances with high, medium, and low settings for that feature. Neural network based synthesis differs from HMM-based synthesis in that NN synthesis does not make use of decision trees. Instead, the frontend features are converted into a binary sequence by way of the questions file, corresponding to "yes" and "no" answers for each question. Pairwise preference results for Merlin subset voices versus the baseline are presented in Table 5, with the best setting for each feature in bold, and results significantly better than the baseline underlined.

We see that a number of voices are rated as more natural than the baseline, with one significant preference: the voice with mean f0 level labeled at the frontend, and test utterances synthesized with the "lo" setting. Although our best Merlin voice is not produced using the same features as our best HTS voice, and although the best setting for each feature is not the same across training methods, we do observe that this frontend-labeling approach can produce significantly more natural voices regardless of the acoustic model.

Next, we wished to see whether the combination of features with their best settings could lead to greater improvement, as we tried with HTS. We added features one at a time and trained voices using them, and synthesized using the best setting for those features as indicated in Table 5. Since we had a three-way tie between medium standard deviation of f0, low mean energy, and medium duration, we posted tiebreaker HITs on MTurk to decide the order in which to add those features. Results for voices with accumulated features are in Table 6.

Table 5: *Percent preference for Merlin voices trained on data labeled as having high, medium, or low values for features and then synthesized with each of the three settings.*

Feature	hi	med	lo
Mean f0	41.7	53.3	**65.0**
Std. dev f0	51.7	**55.0**	50.0
Mean energy	46.7	48.3	**55.0**
Std. dev energy	**61.7**	50.0	60.0
Speaking rate	**50.0**	41.7	48.3
Articulation	41.7	41.7	**53.3**
Duration	48.3	**55.0**	50.0

Table 6: *Percent preference for Merlin voices trained with labels for multiple features combined*

Features	Preference
Mean f0 (lo) + Std. dev. energy (hi)	53.3
+ Duration (med)	48.3
+ Mean energy (lo)	46.7
+ Std. dev. f0 (med)	56.7
+ Articulation (lo)	35.0
+ Speaking rate (hi)	46.7

We see again that combining the best two features does actually not do as well as using each feature separately, and in fact this time, we generally see a *decrease* in naturalness ratings as we add more features. It is possible that this is a result of overfitting from adding too many new features, or possibly that the different features are interacting in ways that hurt naturalness.

6. Conclusions and Future Work

We have found that for both HMM-based synthesis and neural network based synthesis, that adding *individual* acoustic and prosodic features as new frontend labels can significantly improve voice naturalness, but that combination generally does not help. This raises the question of why this is the case, which will require more investigation into the interaction between these different features. We would also like to try the contextual-feature approach with the actual numerical values rather than discretized high, medium, and low settings, since neural networks allow for this type of input. We would also like to explore whether combining the additional frontend features with the adaptation approach could give further improvements. Finally, as the aim of our work is to build voices using broadcast news data in low-resource languages, we would like to see which of our results generalize to other languages, with the aim of building high-quality, natural-sounding voices for a variety of languages making the best use of found data.

7. Acknowledgements

This work was supported by the National Science Foundation under Grant IIS 1548092.

8. References

[1] J. Matoušek, D. Tihelka, and J. Romportl, "Building of a speech corpus optimised for unit selection tts synthesis," *LREC*, 2008.

[2] H. Zen, K. Tokuda, and A. W. Black, "Statistical parametric speech synthesis," *Speech Communication*, vol. 51, no. 11, pp. 1039–1064, 2009.

[3] J. Yamagishi, Z. Ling, and S. King, "Robustness of HMM-based speech synthesis," *INTERSPEECH*, 2008.

[4] A. Gallardo-Antolín, J. Montero, and S. King, "A comparison of open-source segmentation architectures for dealing with imperfect data from the media in speech synthesis," *INTERSPEECH*, 2014.

[5] A. Stan, O. Watts, Y. Mamiya, M. Giurgiu, R. A. Clark, J. Yamagishi, and S. King, "TUNDRA: a multilingual corpus of found data for TTS research created with light supervision," *INTERSPEECH*, 2013.

[6] A. Chalamandaris, P. Tsiakoulis, S. Karabetsos, and S. Raptis, "Using audio books for training a text-to-speech system," *LREC*, 2014.

[7] V. Wan, J. Latorre, K. Yanagisawa, N. Braunschweiler, L. Chen, M. J. F. Gales, and M. Akamine, "Building HMM-TTS voices on diverse data," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 2, pp. 296–306, 2014.

[8] E. Cooper, Y. Levitan, and J. Hirschberg, "Data selection for naturalness in HMM-based speech synthesis," *Speech Prosody*, 2016.

[9] E. Cooper, A. Chang, Y. Levitan, and J. Hirschberg, "Data selection and adaptation for naturalness in HMM-based speech synthesis," *INTERSPEECH*, 2016.

[10] M. Ostendorf, P. J. Price, and S. Shattuck-Hufnagel, "The Boston University radio news corpus," *Tech. Rep.*, 1995.

[11] J. Yamagishi, K. Onishi, T. Masuko, and T. Kobayashi, "Modeling of various speaking styles and emotions for HMM-based speech synthesis," *EUROSPEECH*, 2003.

[12] H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, A. W. Black, and K. Tokuda, "The HMM-based speech synthesis system (HTS) version 2.0," *6th ISCA Workshop on Speech Synthesis*, 2007.

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

[14] M. Morise, F. Yokomori, and K. Ozawa, "WORLD: A vocoder-based high quality speech synthesis system for real-time applications," *IEICE Transactions on Information and Systems*, 2016.

[15] A. W. Black, P. Taylor, and R. Caley, "The Festival speech synthesis system." [Online]. Available: http://www.festvox.org/festival/

[16] P. Boersma, "Praat, a system for doing phonetics by computer," *Clot International, vol. 5, no. 9-10, pp. 341–345*, 2001.

[17] T. Merritt, J. Latorre, and S. King, "Attributing modelling errors in HMM synthesis by stepping gradually from natural to modelled speech," *ICASSP*, 2015.

[18] O. Watts, G. E. Henter, T. Merritt, Z. Wu, and S. King, "From HMMs to DNNs: Where do the improvements come from?" *ICASSP*, 2016.

[19] X. Wang, S. Takaki, and J. Yamagishi, "A comparative study of the performance of HMM, DNN, and RNN based speech synthesis systems trained on very large speaker-dependent corpora," *9th ISCA Speech Synthesis Workshop*, 2016.

[20] B. Bollepalli, M. Airaksinen, and P. Alku, "Lombard speech synthesis using long short-term memory recurrent neural networks," *ICASSP*, 2017.

[21] P. Swietojanski, J. Li, and S. Renals, "Learning hidden unit contributions for unsupervised acoustic model adaptation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 8, pp. 1450–1463, 2016.