# Towards Pooled-Speaker Concatenative Text-to-Speech

*Ellen M. Eide. Michael A. Picheny*

{eeide,picheny}@us.ibm.com

*IBM T.J. Watson Research Center, Yorktown Heights, NY, 10598, USA*

## Abstract

In this paper we explore the merging of data from various speakers in building a concatenative text-to-speech system. First, we investigate the pooling of data from multiple speakers for building statistical models to predict pitch and duration, and present listening test results which show that the expressiveness of our TTS system is improved using these techniques. Additionally, we describe an experiment in which we merged databases from several speakers to form an enlarged database from which our concatenative text-to-speech system draws segments. We present listening test results which show that pooling data from several speakers yields higher quality synthetic speech in general domains than restricting ourselves to the data from just one speaker in our repertoire.

## 1. Introduction

The quality of concatenative text-to-speech (TTS) systems has increased dramatically over the past several years. The improvements have come both from algorithms, such as those described in Hamza [1], Kim [2], and Eide [3], and also from using improved datasets from which the segments are drawn. Empirical observations have shown that systems drawing from high-quality studio recordings of professional speakers far surpass systems using "in-house" recordings spoken by, for example, speech researchers.

Professional speakers tend to have pitch ranges which are far larger than ordinary speakers; for that reason, large datasets are needed in order to provide a sufficiently rich set of prosodic choices for each context. Unfortunately, though, collecting large datasets from professional speakers in a studio can be costly. Apart from monetary concerns, the ability to incorporate data from new speakers in an existing TTS system has appeal, because recording new material from an existing voice eventually becomes impossible once the speaker is no longer available.

In this paper we consider ways to pool data from several professional speakers together to form a larger dataset than is available for any one of the speakers individually. The flexibility of biasing the output towards any of the voices in the combined dataset allows us to offer several distinct voices using the pooled-data approach.

There is an inherent trade-off between having more data available and having those data come from several speakers, thus potentially blurring any speaker-specific peculiarities we may wish to preserve. This paper attempts to identify conditions under which that trade-off favors pooling. Towards that end, we choose *a priori* one of the speakers as the "target" speaker, and label the remaining speakers as "auxiliary" speakers. The pooled-speaker system will have as its goal producing high-quality synthetic speech which sounds like the target speaker through judicious use of data from that speaker as well as from the auxiliary speakers.

In Section 2, we consider pooling data for prosody modeling. The IBM Expressive Text-to-Speech System [4],[1] uses a decision tree for predicting pitch contours for each syllable to be synthesized, and a separate decision tree for predicting durations for each phone to be synthesized. When pooling data for prosody modeling, we do not normalize the observations used to train the duration model, as all of the speakers in the combined dataset spoke at roughly the same speaking rate. By contrast, we do normalize the pitch observations from each of the training speakers as described in Section 2.1 before building pitch models from the combined normalized data.

At runtime, the target-speaker-specific parameters are used to calculate speaker-specific pitch contours from the speaker-independent pitch models. We show in Section 4.1 that pooling data for training prosody models significantly improves the expressiveness of our system.

In Section 3 we consider pooling segments to form a large database from which the TTS engine may draw. This a bit trickier than pooling data for prosody models, because the normalization step is less straight-forward. In the case of an auxiliary speaker whose voice sounds relatively close to the target speaker, no normalization may be needed. However, in general, speaker-identifying characteristics such as pitch and spectral envelope need to be adjusted to bring the auxiliary speaker close to the target speaker. Ultimately we would like to use state-of-the-art voice morphing techniques such as those described in Kawahara [5], Kain [6], and Ye and Young [7]. However, voice morphing is not the focus of this paper; in order to make our diverse voices more homogeneous we simply used a shift in the average pitch to bring the auxiliary speakers' pitch to that of the target speaker. Even using this simplistic approach we have observed a tendency for listeners to prefer a system drawing from the pooled data of 3 speakers to a system drawing from the single-speaker dataset on general-domain sentences, as discussed in Section 4.2. We expect the pooled-data approach to further outperform the single-speaker approach as the quality of our voice conversion increases.

## 2. Pooled-speaker prosody models

In this section we describe pooling data from several speakers to build pitch and duration models. We expect this approach to be especially advantageous in situations where relatively little data is available from the target speaker for building prosody models. One example of such a case is in our expressive speech synthesis system in which we collect a relatively small amount of data in each of a set of expressive styles and build prosody models separately for each style. In this paper we pool the data representing the "conveying good news" style from 3 speakers, 1 female target speaker and 1 female and 1 male auxiliary speaker, to build a pooled "good news" prosody model. The training of the model to predict pitch is described further in Section 2.1, and the training of the model to predict durations is mentioned in Section 2.2.

All speakers had originally read essentially the same script. Thus, we had three times the amount of data available than would be available for speaker-specific models, but the number of observed contexts was essentially unchanged.

We show in Section 4.1 that, at least in the case of conveying good news, listeners prefer the output of the TTS system using the pooled model to that of the system using a model built only from the target speaker's data. We expect that trend to persist, independent of the particular expressive style being examined. We further expect the trend to persist for the neutral case, but perhaps to a smaller degree given that we have roughly eight times the amount of data for the neutral style than is available for any of the other expressive styles.

### 2.1. Pooled-speaker pitch models

The form of the pitch model remains unchanged from our speaker-specific system. We use a decision tree with features derived from the text such as lexical stress, distances from phrase boundaries, etc. We predict one target pitch vector per syllable. The target pitch vector specifies the desired pitch at three points, corresponding to the beginning, middle, and end of the syllable's sonorant region. Although the form of the pitch model is the same in the pooled-speaker and the speaker-specific systems, the observation in each of these cases is different. Rather than modeling the pitch (in the log domain) as before, in the speaker-pooled model we subtract from the pitch the mean of the pitch for the speaker from whom the observation came. Finally, we divide by the standard deviation of the pitch for that speaker. Thus, our normalized observation is $(p-\mu_i)/\sigma_i$ where $p$ is the log pitch, $\mu_i$ is the mean of the log pitch for speaker $i$, and $\sigma_i$ is the standard deviation of the log pitch for speaker $i$.

The new, pooled-speaker decision tree for estimating pitch has approximately 500 leaves, whereas each speaker-specific tree has around 200 leaves. The mean of the normalized observations mapping to leaf $j$ form the prediction vector $x_j$ for that leaf.

At run-time, features are assembled as usual from the text and dropped down the pooled-speaker decision tree. The prediction vector of the appropriate leaf, $x_n$, is then un-normalized using the target-speaker's pitch parameters, $\sigma_T$ and $\mu_T$, to form the pitch target values for that syllable. The $i^{th}$ component of the target vector for a given syllable is given by:

$$p_i = x_{n\,i}\ \sigma_T + \mu_T$$

for $i=\{0,1,2\}$, corresponding to the pitch at the beginning, middle, and end of the sonorant region of the syllable.

### 2.2. Pooled-speaker duration models

All of the professional speakers who contributed data to the speaker-pooled prosody models spoke at roughly the same speaking rate. Thus, in this experiment we did not need to normalize the duration observations before pooling them. However, had we incorporated data spoken at a substantially different rate, we could normalize the durations by dividing by the speaking rate, pool, and then un-normalize at run-time by multiplying the observation by the target speaking rate.

The decision tree for predicting durations from pooled data has approximately 1300 leaves whereas the decision tree for predicting durations from speaker-specific data typically has about 900 leaves.

## 3. Pooled datasets for segment selection

Pooling segments from several speakers in order to form an enlarged dataset for concatenative TTS potentially requires some signal processing to unify originally different pitch and formant ranges. In our experiment, we pooled three professional female speakers, one of whom we identified *a priori* as the target voice. The average pitch of the target voice was 226 Hertz. One of the auxiliary female speakers' pitch and formant positions were fairly similar to the target speaker; we added the segments from this speaker to the pooled database without processing. The third speaker had an average pitch of 168 Hertz, markedly lower than that of the target speaker.

As the focus of this paper is on pooling data from various speakers for concatenative TTS rather than on the details of voice morphing, we chose a commercially-available third party software [8] to process the database of this speaker, adjusting the average pitch to match that of the target speaker. The software does not allow independent control of formants and pitch; having that capability, as well as the ability to process other aspects of the waveform such as breathiness and glottal formant, would enhance the perceived match between the auxiliary and target speakers. The field of voice conversion is rapidly developing; using advanced techniques would undoubtedly help to improve the quality of our pooled-speaker synthesis.

The pitch-adjusted data from this third speaker were then pooled with the data from the other two speakers to form a dataset with approximately three times the amount of data in the target-speaker-specific database, although the number of triphone contexts remained approximately constant because the speakers all read essentially the same script.

### 3.1. Building the Pooled-speaker Database

The process of building the pooled dataset follows the framework developed for the IBM Expressive Speech System described more fully in Hamza [1]. In summary, each segment in the database is labeled by an attribute vector carrying information about that segment. One element of the attribute vector is the identity of the speaker who originally spoke that segment. During synthesis, the input, which is in the form of an extended SSML document, is processed by an XML parser. The extended SSML tags are used to form a target attribute vector, analogous to the one used in the voice-dataset-building process to label the speech segments. In this case, one element of the target attribute vector is the identity of the target speaker. Another element may be the expressive style, say "conveying good news," "conveying bad news," "asking a question," or "neutral" as was considered in Eide [4]. An attribute cost function $C(t,o)$ penalizes the use of a speech segment labeled with attribute vector $o$ when the target is labeled by attribute vector $t$. A cost matrix $C_i$ is defined by hand for each element $i$ in the attribute vector. An example of such a matrix is shown below for the speaker element.

| | Speaker 1 | Speaker 2 | Speaker 3 |
|---|---|---|---|
| Speaker 1 | 0 | 0.2 | 0.5 |
| Speaker 2 | 0.1 | 0 | 0.5 |
| Speaker 3 | 0.3 | 0.3 | 0 |

*Table 1: Cost matrix for "speaker" element of attribute vector. Columns are target speaker; rows are segment source speaker.*

The matrix specifies, for example, that the cost of using a segment from Speaker 2 when Speaker 3 is the target is 0.5. Asymmetries in the matrix may arise because of different sizes of datasets. If one speaker has a very large dataset compared to another speaker, it may make sense to penalize more heavily the use of segments from the smaller dataset when the speaker with the large dataset is the target, and to penalize less heavily the use of segments from the large dataset when the speaker corresponding to the small dataset is the target.

## 4. Results

### 4.1. Pooled-Speaker Expressive Prosody Models

In this section we report the results of an experiment in which we compared the quality of communicating good news to listeners using pooled-speaker vs. speaker-specific expressive prosody models. The speaker-specific models were built from roughly 1,000 good news sentences read by the target speaker. The speaker-pooled models included data from the target speaker as well as one male and one additional female speaker. All segments were spoken by the target speaker. Listeners were presented with 29 sentences from each of the two systems and were asked to rate the quality of the system in delivering the message on a scale of 1 (poor) to 5 (excellent). Ten male and ten female native-US-English speaking listeners participated. Results are shown in Table 2, and are significant at the $p < 0.05$ level.

| Prosody Model | MOS |
|---|---|
| Speaker-specific | 3.56 |
| Speaker-pooled | 3.72 |

*Table 2: MOS Results for Speaker-specific vs. Speaker-pooled prosody models in conveying good news.*

### 4.2. Pooled-Speaker Segments

In this section we report the results of a listening test in which we compare the quality of synthesis generated from the target-speaker-specific database with the quality of synthesis generated from the pooled databases of three professional female speakers as described in Section 3. Target-speaker-specific prosody models were used for both the target-speaker-specific and the pooled segment cases.

On average, about 50% of the segments chosen were originally from the target speaker, about 40% of the segments were from the (unprocessed) first auxiliary speaker, and about 10% of the segments were from the (processed) second auxiliary speaker. These percentages can be adjusted by tuning the contribution to the cost function of the penalty for a mismatch between the segment speaker and the target speaker given in Table 1. However, even with a very low speaker substitution cost, the spectral-continuity component of the segment-selection cost function works to ensure that large spectral mismatches are not spliced together. Shown in Figure 1 is a spectrogram resulting from the pooled-speaker system. No obvious spectral discontinuities are observed, even though all 3 speakers are represented in this waveform.

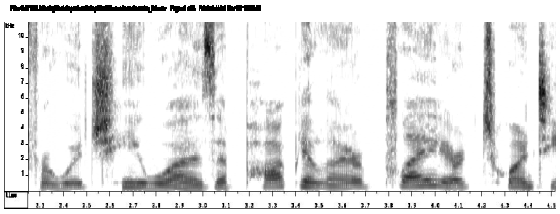

*Figure 1: Spectrogram of pooled-speaker TTS*

We ran a listening test consisting of ten male and ten female native U.S. English speakers rating the quality of output from the target-speaker-specific and the pooled-speaker systems on a scale of 1 (poor) to 5 (excellent). Twenty-five sentences from each system were presented to the listeners; fifteen of the sentences were "in-domain" in that they were about topics which were represented in the speakers' scripts, and ten of the sentences were "general-domain" in that their subject matter was not specifically covered in the speakers' scripts. We present the results for these two cases separately, in Tables 3 and 4, respectively.

| Segments | MOS |
|---|---|
| Speaker-specific | 3.50 |
| Speaker-pooled | 3.45 |

*Table 3: MOS Results for Speaker-specific vs. Speaker-pooled segments on in-domain sentences.*

On the in-domain sentences, listeners showed a slight preference for the speaker-specific models. For these sentences the speaker-specific database contains a rich set of segments from which to choose; the speaker-specific data are adequate for producing good quality output and listeners tended to prefer the homogeneity of the voice to the small increase in prosodic richness afforded by the speaker-pooled dataset.

On the other hand, for the general-domain sentences, listeners demonstrated a preference for the pooled-speaker system over the speaker-specific one, as indicated in Table 4. In this case the improved spectral smoothness and prosodic structure afforded by the increase in dataset size outweighed the loss in homogeneity by pooling three voices. Interestingly, in casual conversation by the participants after the tests, nobody remarked that he/she had perceived the sentence as being uttered by more than one voice.

| Segments | MOS |
|---|---|
| Speaker-specific | 3.06 |
| Speaker-pooled | 3.16 |

*Table 4: MOS Results for Speaker-specific vs. Speaker-pooled segments on general-domain sentences.*

## 5. Discussion

In this paper we have examined the use of data from several speakers in building a concatenative text-to-speech system. We explored separately the sharing of data for building prosody models and the sharing of data for selecting segments for concatenation. Presumably using each of these techniques together would result in larger improvements over the speaker-specific methods than either technique separately.

For applications in which only a very small sample of a target voice is available, we expect that the pooled-data approach to prosody modeling will be very useful as a part of a system employing voice morphing, because we need only estimate a mean and standard deviation for the target speaker in order to generate a well-estimated prosody model.

The quality of the speech output in the case of pooled segments relies on high-quality of the signal processing to convert auxiliary voices to sound as much as possible like the target speaker. The inherent tradeoff between the amount of data available and the quality of the morphing and voice match will increasingly balance at larger and larger amounts of morphing as that technology matures. Thus, we expect the improvements from pooling techniques to outperform speaker-specific methods by ever-increasing amounts.

## 6. Acknowledgements

Thanks to Andy Aaron of IBM for discussions of commercially available software to alter pitch, and to Allen Delmar of IBM for running the listening tests reported herein.

## 7. References

[1] Hamza, W. et al. "The IBM Expressive Speech Synthesis System." Proceedings ICSLP, 2004, Jeju Island, Korea.

[2] Kim, Y., A. Syrdal, and M. Jilka. "Improving TTS by Higher Agreement Between Predicted Versus Observed Pronunciations." Proceedings of the 5th ISCA Speech Synthesis Workshop, Pittsburgh, PA, USA, June 14-16, 2004.

[3] Eide, E. et al. "Recent Improvements to the IBM Trainable Speech Synthesis System." Proc. ICASSP 2003. Hong Kong, China.

[4] Eide, E. et al. "A Corpus-based Approach to <ahem/> Expressive Speech Synthesis." Proceedings of the 5th ISCA Speech Synthesis Workshop, Pittsburgh, PA, USA, June 14-16, 2004.

[5] Kawahara, H. "Restructuring speech representations using a pitch adaptive time-frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds," Speech Communication 27, pp. 187-207, 1999.

[6] Kain, A. "High Resolution Voice Transformation." PhD thesis, OGI School of Science and Engineering at Oregon Health and Science University. 2001.

[7] Ye, H. and S. Young. "High Quality Voice Morphing," Proceedings ICASSP, Montreal, Canada, 2004.

[8] Adobe Audition 1.5 Software . http://www.adobe.com/products/audition/main.html