Database Pruning for Indian Language Unit Selection Synthesizers

Veera Raghavendra
LTRC, IIIT-Hyderabad
Hyderabad, India
raghavendra@iiit.ac.in

Kishore Prahallad
LTRC, IIIT-Hyderabad
Hyderabad, India
kishore@iiit.ac.in

Abstract

The size of unit selection speech synthesis is between few hundred of MBs to GBs. Such a huge database requires a large memory size and slows down the computational speed. It also causes too much hindrance to download and install in ordinary machines. To some, it may look old-fashioned to worry about size and speed of a software application. With ever-increasing CPU speed and disk sizes growing continuously, many have forgotten what it is like to be restricted in memory and computational complexity. However, to those wishing to make speech applications ubiquitous, it quickly becomes clear that not all applications are deployed in resource-rich environments, with lots of CPU cycles to burn and large amount of memory and storage. In this paper we propose three methods for pruning large unit selection databases to be able to deploy in practical applications. All these techniques are evaluated using objective measures.

1 Introduction

The ability to produce high quality synthetic speech is quickly followed by the demand for high quality speech synthesis on range of small devices: mobile telephones, embedded systems, and hands-free devices, which pose interesting challenges for modern synthesizers - especially those using concatenative synthesis methods. Though the memories of such devices are expanding day-to-day, amount of space allotted to synthesis is very low which may be around 20 MB. Most of the memory space is used for storing personal audio and video files. In unit selection speech synthesis, the sentence is synthesized by joining pre-recorded speech segments. A large scale database with various spectral and prosodic instances of each unit is created to improve the naturalness of speech synthesis. The quality of synthetic speech is proportional to database size. Now-a-days the size of unit selection speech synthesizer is around 2 GB. Such a huge database requires a large memory space and also higher computational power. It also poses too much hindrance to download and install, especially for people in third-world countries using machines with limited storage and CPU power. Thus the issue here is to come-up with a method of reducing the speech database with minimal loss of naturalness and intelligibility.

2 Approaches for database pruning

Several approaches for reducing the size of unit selection voices have been proposed. The approach described in (Jerome R. Bellegarda, 2008; Jerome R. Bellegarda, 2007) addresses that two kinds of units have to be removed to prune the database. The first approach is to remove the spurious units, known as “outliers“, which may have been caused by mislabeling. The second approach is to remove those units which have similar characteristics between instances for a given unit. The general idea is to cluster together units that are “similar“ and compare units from each cluster with its corresponding cluster center. Pruning is then achieved by removing those instances that are “farthest away“ from the cluster center.

In (Zhao, Y. and Chu, M. and Peng, H. and Eric Chang., 2004), it combines two methods for reducing the database size. The first method is removing the outliers which are occurred because of mistakes in unit boundary alignment or break-

Proceedings of ICON-2009: 7th International Conference on Natural Language Processing, Macmillan Publishers, India. Also accessible from http://ltrc.iiit.ac.in/proceedings/ICON-2009.

indices labeling. Average f_0 and duration factors are used to remove such kind of outliers. To remove outliers, phonetically similar instances of the unit are clustered and some threshold is defined for average f_0 and duration. The instances which are above the threshold are removed from cluster, which means that instances are away from the cluster center. The leftover instances have similar prosodic features. The second method is identifying the redundant instances that might be generalized as less frequently used instances or less important than that of the frequently used ones. The importance of an instance can be measured by its contribution to synthetic speech, defined as the usage frequency of the instances divided by the accumulative usage frequency of all instances after synthesizing a large amount of text.

Kim *et.al.,* presented a weighted vector quantization (WVQ) method that prunes the least important instances. 50% reduction rate is reached without significant distortions. In (Black, A. W. and Taylor, P. A., 1997), each unit is represented as a sequence of frames, or vectors of Mel-Cepstral Coefficients (MCEP), and decision tree clustering proceeds based on questions concerning prosodic and phonetic context; units are then assessed based on their frame based distance to each cluster center.

In (Hon, H. and Acero, A. and Huang, X. and Liu, J. and Plumpe, M., 1998), similar instances of the unit are clustered using decision trees. Two approaches are being used for database pruning. In the first approach an instance is selected randomly from unit cluster. Such approach produces large glitches for some concatenations. To avoid this problem, HMM scores would be calculated for each instance in the cluster using Viterbi alignment. By choosing the unit instance with highest HMM score to represent the cluster, this approach is able to produce good concatenation quality. In the second approach, instead of selecting one highest HMM score, top 10 highest HMM score units are selected. The resulting TTS was able to produce good quality synthesis.

Flite (Black, A. and Lenzo, K., 2001) is a small fast run-time synthesis engine developed at CMU and primarily designed for embedded machines and/or large servers. Flite is designed as an alternative synthesis engine to Festival (Black et al., 1998) for voices built using the FestVox (Black and Lenzo, 2000) suite of voice building tools. Flite uses diphone concatenative technique for synthesizing speech. The database of units that are to be concatenated is represented in terms of LPC coefficients.

In (Chazan, D. and Hoory, R. and Kons, Z. and Silberstein, D. and Sorin, A., 2002), it is proposed to compress the database using vector quantization (VQ) for reducing the database size. The speech parameters to be compressed include the MCEP feature vectors and the degree of voicing for the synthetic phase. MCEP features are quantized using split VQ, while the degree of voicing is coded with scalar quantization. MCEPs, pitch and degree of voicing are extracted for every 10ms frame from the speech signal and features are coded for every 20msec. In the interleaved frame only an interpolation factor is coded. During synthesis pitch, energy and duration are predicted and MCEPs are estimated using VQ. The speech is reconstructed using a novel technique (Chazan, D. and Hoory, R. and Cohen, G. and M. Zibulski,, 2000) from the given MCEP features and pitch.

In all the above approaches more than one unit variation is preserved to synthesize the speech. It again involves target cost to select the best unit. To avoid this problem, we are investing towards selection of one best unit. The question is - what is the criteria for selecting the most suitable unit out of the several instances to form the scaled down database. We experimented with several alternatives for the most suitable unit going all the way from defining it as a neutral/average unit to an optimal unit.

The rest of the paper is organized as follows. Section 3 gives the database details used in the experiment. Section 4 discusses the database pruning techniques. Section 5 gives the evaluation of the experiments and Section 6 gives the summary of the paper.

3 Speech Database Used

The quality of the unit selection voices depends to a large extent on the variability and availability of representative units. It is crucial to design a corpus that covers all speech units and most of their variations in a feasible size. The speech databases used for Telugu, Hindi, and Tamil are recorded by 3 different female speakers. The details of the corpus are given in the Table 1. All sentences are recorded in a professional studio and the sentences are read in a relaxed reading style, which is between “for-

mal reading style" and "free talking style", at moderate speaking rate. Recordings are performed in a soundproof room with close-talking microphone. The speech database has been phonetically labeled using Ergodic hidden Markov models (EHMM) (Prahallad et al., 2006), which is well tuned to automatic labeling for building voices in Festvox (Black and Lenzo, 2000) framework. Using this tool, context-independent models with two Gaussians per state are generated using 13 Mel Frequency Cepstral Coefficients (MFCCs). Once the phone labels are obtained, they are extended to get the syllable boundaries which will be used in the syllable based synthesis. These syllables are extracted from global syllable database (Raghavendra et al., 2008a).

Table 1: *Language database details.*

Language	No.Of. Sentences	No.Of. Words	Unique Words
Telugu	1631	27303	8026
Hindi	585	14398	4415
Tamil	2392	33945	7817

4 Experiments for selecting best unit

We have investigated three different approaches for selecting the best suitable unit. In the following sub sections we describe how to build a scaled down database using single instance for each unit type.

4.1 Average and Euclidean distance method

Assume that for each unit type of interest say ($syllable$), m instances are present in the database. First step is to gather these m instances, and divide into four categories based on positional context in the given word (Samuel, T. and Rao, M.N. and Murthy, H.A. and Ramalingam, C.S., 2006). The categories are listed as below.

- Word syllable (wsyllable) - a mono syllable word.
- Initial syllable (bsyllable) - 1^{st} syllable of the word.
- Middle syllable (msyllable) - other than 1^{st} and last syllable of the word.
- Ending syllable (esyllable) - last syllable of the word.

This categorization of syllable ensures that syllable is chosen based on its position. Such selection of unit based on appropriate position captures the stress information and pauses at word boundaries and improves the quality of synthesis. Table 2 gives the details of unique and total number of syllables for each category. These syllables are generated using global syllable set.

Table 2: *Database details of the each category.* $bsyllables$ *denote the initial syllable,* $msyllables$ *denote the middle syllable,* $esyllables$ *denote the ending syllable and* $wsyllables$ *denote the word syllable.*

Category	Unique Syllables	Total Syllables
bsyllables	715	46511
msyllables	1790	56878
esyllables	3035	46511
wsyllables	788	5484
Total	6328	155384

In second step, acoustical features *energy, fundamental frequency (f_0),* and *duration* are extracted for each instance of the unit. Energy and f_0 are analyzed for each frame with 10ms frame size and 5ms frame shift and averaged over all the frames of the syllable duration. Finally a $M * N$ matrix is constructed as follows

$$A_{m*n} = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \vdots & \vdots \\ a_{m1} & \dots & a_{mn} \end{bmatrix}$$

where m is the number of instances and n is the number of features (energy, f_0 and duration). However, the range of energy, f_0 and duration values are different. To bring all the values in particular range, each value is normalized between 0 and 1 with $maximum$ value of each column.

Once normalized matrix is obtained, we attempt to select a unit from multiple instances of the unit present in the database, such that the selected unit is prosodically neutral with minimal influence of its context. The criteria for selecting a neutral unit is based on the hypothesis that it would join together pretty well with each other though the speech thus produced may not have naturalness. To select this neutral unit, mean is calculated over all the feature vectors and considered as the local threshold. Euclidean Distance is performed between the instance feature vector and mean vector. A statistically consistent unit is selected by choos-

ing an instance which is closest to the mean vector as shown in equation 3.

$$\mu = [\sum_{i=1}^{m} a_{i1}/m, \sum_{i=1}^{m} a_{i2}/m, \ldots, \sum_{i=1}^{m} a_{in}/m] \quad (1)$$

$$d_m = \sqrt{\sum_{j=1}^{n} (A_{mj} - \mu_j)^2} \quad (2)$$

where μ is the mean vector

$$D = argmin_m(d_m) \quad (3)$$

4.2 Selecting a neutral unit using principle component analysis (PCA)

PCA is a tool in data analysis. It is a non-parametric method for extracting relevant information from the data (Smith, 2002) by projecting the data onto a lower dimension to reveal the hidden structure that underlie it. This technique is generally used in various fields including image compression and speech recognition. In this section we will see how to use PCA technique in the context of pruning in speech synthesis.

Assume that for the unit type of interest, M instances are present in the database. First step is to gather these M instances, extracting the acoustic features of each instance like duration, energy, f_0 and MCEPs for each frame with 10ms frame size and 5ms frame shift. Later join together all the frame features of syllable segment. If N denotes the maximum number of components of the whole instances, we then zero-pad all units to N, as necessary. The outcome is MXN matrix W with elements w_{ij}, where each row w_i corresponds to a particular instance, and each column corresponds to a slice of feature. The dimensionality of the matrix depends on each unit type and it would be in 100s or 1000s. Using PCA, N dimensional data can be projected onto L dimension and is done as follows.

$$A_i = (w_i - \mu)\phi^T$$

Where A_i is the lower dimensional vector for each instance, μ is the mean over all the instances of a unit, ϕ^T is transpose of (LXN) eigenvector matrix. The number of eigenvectors is selected using following formula.

$$[\frac{\sum_{i=1}^{L} \lambda_i}{\sum_{i=1}^{N} \lambda_i}] * 100 \geq 99\%$$

Where λ is the descending order of eigenvalues of the matrix W. The size of the reduced matrix is (MXL). To select a neutral unit, mean is calculated over all the feature vectors and considered as the local threshold. Euclidean Distance is performed between the instance feature vector and mean vector. A statistically consistent unit is selected by choosing an instance which is closer to the mean vector as shown in equation 6.

$$\mu = [\sum_{i=1}^{m} a_{i1}/m, \sum_{i=1}^{m} a_{i2}/m, \ldots, \sum_{i=1}^{m} a_{in}/m] \quad (4)$$

$$d_m = \sqrt{\sum_{j=1}^{n} (A_{mj} - \mu_j)^2} \quad (5)$$

where μ is the mean vector.

$$D = argmin_m(d_m) \quad (6)$$

4.3 Database pruning using dynamic time warping

Dynamic time warping (DTW) is a technique that finds the optimal alignment between two time series (reference and input) where one time series is "warped" non-linearly by stretching or shrinking it along its time axis. This warping between two time series can then be used to find corresponding regions between the two time series or to determine the similarity between the two time series. This is roughly equivalent to the problem of finding the minimum distance in the trellis between two time series. Associated with every pair (i, j) is a distance $d(i, j)$ between two vectors x_i and y_j to find optimal path between starting point $(1, 1)$ to (N, M) and identify the one that has the minimum distance. Since there are M possible moves for each step from left to right, all the paths from $(1, 1)$ to (N, M) will be exponential. DTW principle can drastically reduce the amount of computation by avoiding the enumeration of sequences that cannot possibly be optimal. Since the same optimal path after each step must be based on the previous step, the minimum distance $D(i, j)$ must satisfy the following equation.

$$D(i, j) = min_k[D(i - 1, k), d(k, j)] \quad (7)$$

Equation 7 indicates you only need to consider and keep the best move for each pair although

there are M possible moves. The recursion allows the optimal path search to be conducted incrementally from left to right. In essence, DTW delegates the solution recursively to its own sub-problem. The computation proceeds from the small sub-problem $D(i-1,k)$ to the larger sub-problem $D(i,j)$. We can identify the optimal match y_j with respect to x_i and save the index in a back pointer table $B(i,j)$ as we move forward. The optimal path can be back traced after the optimal path is identified. DTW is often used in speech recognition to determine if two waveforms represent the same spoken phrase. In a speech waveform, the duration of each spoken sound and the interval between sounds are permitted to vary, but the overall speech waveforms must be similar. In our work we use DTW for pruning the database in speech synthesis. This is done by generating an average/neutral unit from all the instances of each unit type using DTW.

4.3.1 Selection of statistically consistent unit

Figure 1 shows the different length of instances for the syllable maa. Using DTW, a single averaged instance can be created from these multiple instances of different lengths. However, such an approach needs an instance to be chosen as reference instance or model unit. One solution is to consider the neutral unit obtained from Section 4.1 as model unit and compute the optimal alignment between each instance and the model unit.

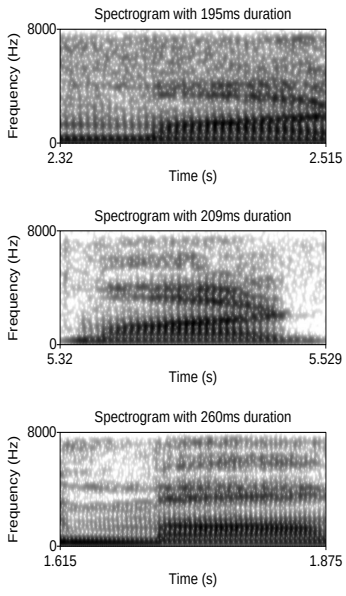

Figure 1: Spectrogram representation for the begin syllables maa with different durations.

To align every instance with a model, 25 dimensional MCEP feature frame and f_0 are used. Euclidean Distance measure is used to find the distance between frames. Following algorithm gives the detailed procedure.

1. Pick the model instance.
2. Take the first instance frames and align to the model frames.
3. repeat step 2 for each another instance.
4. create a new instance by averaging together all frames that align together.

The average instance obtained as a result of above process is considered as a pruned unit.

5 Evaluation

Syllable based synthesizer suffers from coverage of syllable units. Hence, we have used syllable approximate matching algorithm when required syllable is not available in the database (Raghavendra et al., 2008b). To reduce the number of substitution/deletions in the sentence, global syllable set (Raghavendra et al., 2008a) approach was also employed. This set was prepared by combing syllables from Telugu, Hindi and Tamil. The proposed approaches reduced the database size from 2 Gigabytes to around 51 Megabytes. Here the reduction ratio from original to reduced database size was 39:1. Resultant speech database contains only one unit for each category irrespective of the context.

5.1 Acoustical observation

Figure 2 shows the spectrograms for a single phrase synthesized using four different techniques as follows. Figure 2(a) shows the spectrogram of the natural recording of a phrase . Figure 2(b) shows the spectrogram of the phrase synthesized using average technique. Figure 2(c) and 2(d) shows the synthesized speech using PCA and DTW techniques respectively. It could be observed that the average, PCA and DTW technique do preserve the required speech characteristics while just using a single instance of each unit.

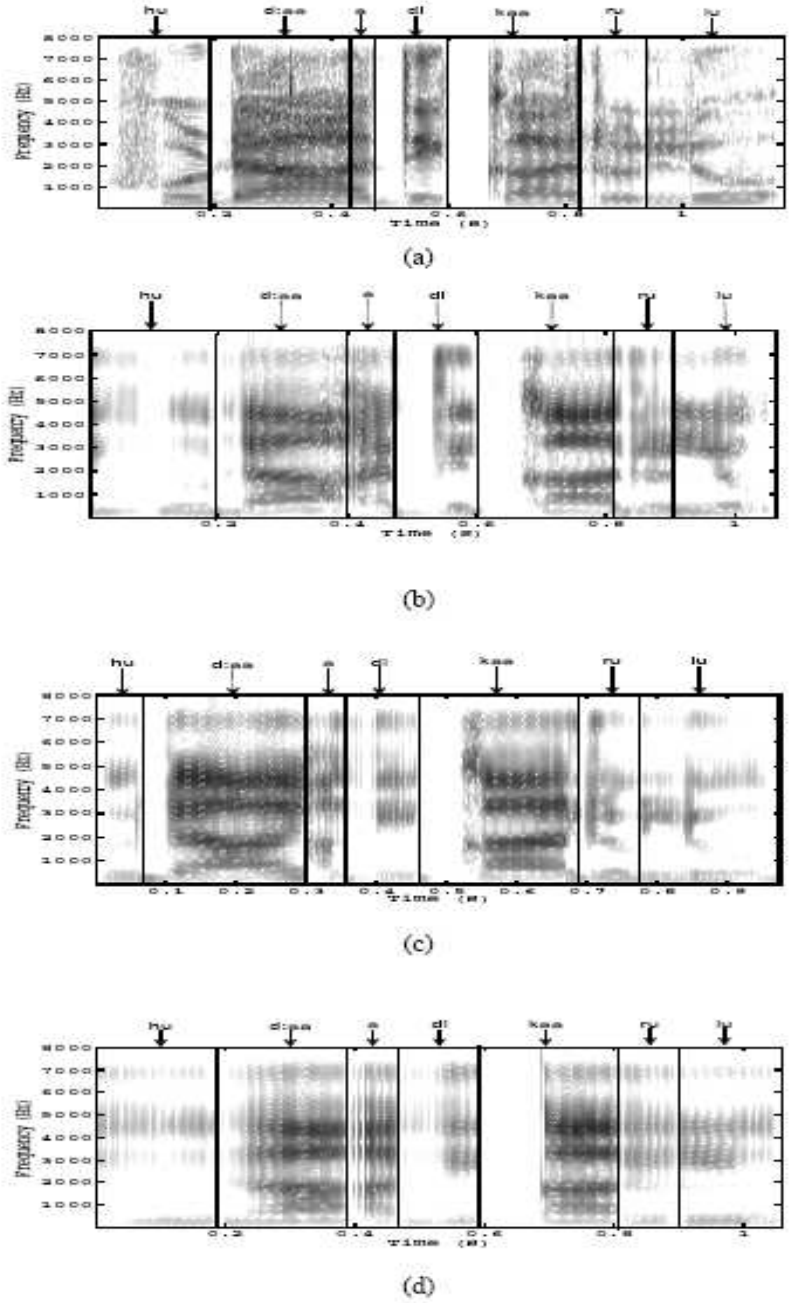

Figure 2: Spectrograms for the phrase *hud:aa adikaarulu* (a) original (b) synthesized from average technique (c) synthesized from PCA technique and (d) synthesized from DTW technique.

Table 3: *MCD scores for global syllable set, average, PCA and DTW techniques.*

	Global syllable set	Average	PCA	DTW
MCD	6.689	7.441	7.478	7.28

5.2 Objective evaluation

Mel cepstral distortion (MCD) is an objective error measure used to compute cepstral distortion between original and the synthesized MCEPs. Lower the MCD value the better is the synthesized speech. MCD is essentially a weighted Euclidean distance defined as

$$MCD = (10/ln10) * \sqrt{2 * \sum_{i=1}^{25} (mc_i^t - mc_i^e)^2} \quad (8)$$

where mc_i^t and mc_i^e denote the target and the estimated MCEPs, respectively. MCD is used as an objective evaluation of synthesized speech (Black, A., 2006). Informally it is observed in (Black, A., 2006) that a difference of 0.2 MCD makes a difference in the perceptual quality of the synthesized signal and typical values for synthesized speech are in the range of 5 to 8 MCD. To compute MCD, we have taken ten test sentences from Telugu database and synthesized using global syllable set (Raghavendra et al., 2008a), average, PCA and DTW techniques. Here global syllable is used as reference system. Table 3 gives the MCD scores for each technique.

The results shown in Table 3 indicate that the pruning of speech database produce higher MCD values in comparison with the global syllable set. When compared between three techniques, DTW based synthesizer is performing better than average and PCA techniques. One reason might be that the averaging across the frames leads to computation of average unit and could be viewed as an approach towards statistical parametric synthesis (Black, A., 2006). Informal listening studies using pruned databases showed that the quality of the synthesized speech using average, PCA and DTW produce intelligible speech, but listeners considered the quality of speech was degraded in comparison with global syllable set. However, it should be noted in the context that pruned synthesizer use a single instance of each unit type, where as global syllable set stores all possible instances for each unit. Hence, it could be considered as a trade-off between the synthesis quality and size of the database.

6 Summary

This paper discusses need for database pruning and approaches followed previously. All the available techniques preserve more than one unit variation for a unit type during synthesis. To reduce the database furthermore, we have proposed three techniques. The first technique uses simple average and Euclidean distance method, the second technique uses PCA and the third technique uses DTW. Evaluations on these three techniques showed that neutral units selected by average, PCA and DTW techniques do preserve the required speech characteristics while just using a single instance of each unit. Objective evaluation showed that there is degradation in the database pruning compared to global syllable set and also that DTW technique is better than other two techniques.

References

Black, A. and Lenzo, K. 2001. Flite: a small fast runtime synthesis engine. In *ISCA, 4th Speech Synthesis Workshop*, pages 157–162.

Black, A. W. and Taylor, P. A. 1997. Automatically clustering similar units for units selection in speech synthesis. In *Proceedings of Eurospeech*, pages 601–604.

Black, A. 2006. CLUSTERGEN: a statistical parametric synthesizer using trajectory modeling. In *Proceedings of Interspeech*, pages 1762–1765.

A.W. Black and K. Lenzo. 2000. Building voices in the Festival speech synthesis system.

A.W. Black, P. Taylor, and R. Caley. 1998. The Festival speech synthesis system.

Chazan, D. and Hoory, R. and Cohen, G. and M. Zibulski,. 2000. Speech reconstruction from mel frequency cepstral coefficients and pitch. In *Proceedings of ICASSP*.

Chazan, D. and Hoory, R. and Kons, Z. and Silberstein, D. and Sorin, A. 2002. Reducing the footprint of the IBM trainable speech synthesis system. In *Proceedings of ICSLP*.

Hon, H. and Acero, A. and Huang, X. and Liu, J. and Plumpe, M. 1998. Automatic generation of synthesis units for trainable text-to-speech systems. In *Proceedings of ICASSP*, volume 1, pages 293–296.

Jerome R. Bellegarda. 2007. LSM-based unit pruning for concatenative speech synthesis. In *Proceedings of ICASSP*, pages IV–521–IV–524, April.

Jerome R. Bellegarda. 2008. Unit-centric feature mapping for inventory pruning in unit selection text-to-speech synthesis. *in IEEE Transaction on Audio, Speech and Language Processing*, 16(1):74–82, January.

K. Prahallad, A.W. Black, and R. Mosur. 2006. Sub-phonetic modeling for capturing pronunciation variations for conversational speech synthesis. In *Proceedings of ICASSP*, France.

E. V. Raghavendra, D. Srinivas, B. Yegnanarayana, A.W. Black, and K. Prahallad. 2008a. Global syllable set for speech synthesis in indian languages. In *Proceedings of IEEE workshop on Spoken Language Technologies.*, Goa, India, December.

E. V. Raghavendra, B. Yegnanarayana, and K. Prahallad. 2008b. Speech synthesis using approximate matching of syllables. In *Proceedings of IEEE workshop on Spoken Language Technologies.*, Goa, India, December.

Samuel, T. and Rao, M.N. and Murthy, H.A. and Ramalingam, C.S. 2006. Natural sounding TTS based on syllable-like units. In *Proceedings of EUSIPCO*, Florence, Italy, September.

L.I. Smith. 2002. A tutorial on principle component analysis. Technical report.

Zhao, Y. and Chu, M. and Peng, H. and Eric Chang. 2004. Custom-tailoring TTS voice font-keeping the naturalness when reducing database size. In *Proceedings of Eurospeech*, pages 2957–2960, Geneva.