

# Automatic Pruning of Unit Selection Speech Databases for Synthesis without loss of Naturalness

Rohit Kumar and S. P. Kishore

Language Technologies Research Center  
International Institute of Information Technology, Hyderabad INDIA  
{ rohit, kishore } @ iit.net

## Abstract

In the paper we present our experiments with automatic pruning of speech databases created by us for Unit Selection based speech synthesis systems. Several algorithms have been attempted and perceptually evaluated. An optimal size of speech database has been reached where lose of naturalness due to unit pruning is not perceptible.

## 1. Introduction

Current state of the art text – to – speech systems [1] are based on the data driven approaches [2] wherein speech is synthesized by concatenation of suitable basic units selected from a huge annotated corpus of recorded speech containing multiple instances of the basic units of speech providing wide coverage of different phonetic and prosodic properties. The selection is made on the basis of minimization of target and concatenation cost functions [3][4].

Generally the size of speech corpus with sufficiently wide coverage to produce natural sounding speech is huge (200-300 MB per voice). Hence, size of these databases restricts the use of such systems for several applications like Palmtops / Mobile Handsets based interactive systems.

Several attempts [5][6][7] have been made to reduce the size of speech databases by pruning off the unit instances that do not contribute to the desired coverage of the database. There is scope for this kind of pruning as several instances will be redundant in terms of prosodic and phonetic features. Further some instances (called dirty units) have very unusual features that are too deviant from the usual values for such units and these instances never get selected because of the high costs they carry while selection. These instances can also be pruned out of the database.

We have multiple aims for experimenting with Unit Pruning. First - to come up with a Low Memory Device Synthesizer that is based on Data Driven approaches but is of sufficiently small size to fit into a PDA. Second - to automate the mechanism to bring up any desired size of speech database depending upon the requirements and system resources affordable, ofcourse with a corresponding level of naturalness. Finally - to develop an optimal sized speech database without loss of naturalness.

The organization of this paper is in correspondence with our aims. In section 2 we describe the attempts at developing a low memory device synthesizer. The idea of neutral and optimal units is introduced. The unit pruning approach is discussed in Section 3. Also it discusses heuristics related to our second aim. Perceptual tests and analysis to approximate an optimal database are described in section 4.

## 2. Low Memory Device Synthesizer

The motivation to develop a Low Memory Device Synthesizer (LMDS), which is small and fast enough to fit into small devices, was due to several commercial and non-commercial interests. To extend our approach for developing an Indian Language Speech Synthesizer using Syllable as the basic unit of synthesis [8] to LMDS, we selected one unit for each syllable and phone (the basic units) out of their multiple instances in the database to form a scaled down database. Synthesis was done merely by breaking up the given text to a sequence of available units and then concatenating them. It must be noted here that Unit Selection job is not involved during this synthesis. Rather it has now been shifted to inventory building process.

### 2.1. Criteria for most suitable unit

The question we need to answer here is that what should be the criteria for selecting this one most suitable unit out of the several instances to form the scaled down database. We experimented with several definitions for this most suitable unit going all the way from defining it as a neutral / average unit to an optimal unit.

#### 2.1.1. Selecting the Neutral/Average Units

Initially we attempted to select a unit from the multiple instances of the unit present in the database, such that the selected unit is prosodically neutral with minimal influence of the context. The criteria was based on the hypothesis that a neutral units would join together pretty well with each other though the speech thus produced may not have naturalness. To select these Neutral units, they were defined as Average units i.e. the Unit with prosodic features closest to the average prosodic features of the instances of that type of units. We refer to this selection criteria as the Algorithm A.

Hereby we will use the following notation in this paper. X, Y, Z to represent basic units (syllables and phones).  $A_i$ ,  $B_j$ ,  $C_k$ ... will be used to represent instances of basic units. Prosodic Feature Pitch, Duration and Energy will be represented by P, D, E with appropriate subscripts.

So by the above definition of Neutral/Average Units, ideally for a basic unit X, the neutral unit should have prosodic features  $\langle P_{Ideal}[X], D_{Ideal}[X], E_{Ideal}[X] \rangle$  such that

$$P_{Ideal}[X] = \frac{1}{N} \sum_{i=1}^N P[A_i] \quad \text{Equation (1)}$$

$$D_{Ideal}[X] = \frac{1}{N} \sum_{i=1}^N D[A_i] \quad \text{Equation (2)}$$

$$E_{Ideal}[X] = \frac{1}{N} \sum_{i=1}^N E[A_i] \quad \text{Equation (3)}$$

where  $N$  is the number of instances and  $A_i$  are the various instances respectively of basic unit  $X$  in the database. Let the unit with these ideal prosodic features be referred to as  $A_{Ideal}$ . So the prosodically neutral unit  $A_{Neutral}$  is taken to be unit that has Minimum Prosodic Mismatch with respect to  $A_{Ideal}$ . The mismatch measure used is the Prosodic Matching Function (PMF) described in [8].

Based on Algorithm A, a databases consisting of the neutral instances of each basic unit  $X$  in the database was created and called as the Database A.

Further, we experimented with some heuristic, which we believed would lead to selection of better (more prosodically neutral) units.

*Database B.:* Giving Preference to instances which are preceded and succeeded by either of ‘a’ or ‘aa’ vowels.

*Database C.:* Avoiding instances preceded and succeeded by Semivowels ‘y’, ‘r’, ‘l’, ‘v’.

*Database D.:* Picking up unit instances from word middle position rather than word begin or word end positions.

No significant change in quality of concatenatively synthesized speech was observed using Databases B, C & D.

### 2.1.2. Hand Labeling the Neutral Units

Hand labeling of the Neutral Instances in Database A was attempted. Each of the Neutral Instance were observed using a labeling tool and its boundaries were pruned to remove unwanted regions at the start and end of each Neutral unit where co-articulations effects were observable. If a unit was found to be very unsuitable for use, Algorithm A was used to get the next most neutral instance and hand labeling was done on that. Finally a Database E of hand labeled neutral units was created.

### 2.1.3. Selecting the Optimal Units: Algorithm F

We look at the question that we asked at the start of section 2.1 again. What are the criteria to decide which instance is the best? A related question is: What is desirable characteristic of the best unit? Hypothesis leading to selection of Average/Neutral units for the best units answers this question by proposing that the best units should be prosodically neutral. An alternative proposition for characteristics of best units is that, it should fit as naturally as possible with the other units that may be adjacent to it in a given random text. The idea of optimal units is based on this.

Optimal Unit of a basic unit  $X$  is defined as the one that has minimum cumulative prosodic mismatch with respect to all the units that preceded (or succeeded) the actual instances of  $X$  in a large spoken corpus. Let  $A_i$  be the various instances of  $X$  in the database,  $A_i^{-1}$  be the unit preceding  $A_i$  and  $A_i^{+1}$  by the unit succeeding  $A_i$ . Also let  $P_{Expected}[A_i^{-1}]$ ,  $D_{Expected}[A_i^{-1}]$ ,  $E_{Expected}[A_i^{-1}]$  be the prosodic expectations of  $A_i^{-1}$  for the unit that is going to be selected after it. Hence,

$$\begin{aligned} P_{Expected}[A_i^{-1}] &= P[A_i] \\ D_{Expected}[A_i^{-1}] &= D[A_i] \\ E_{Expected}[A_i^{-1}] &= E[A_i] \end{aligned} \quad \text{Equation ( 4 )}$$

where  $P[A_i]$ ,  $D[A_i]$ ,  $E[A_i]$  are the pitch, duration and energy of  $A_i$ . So if  $K$  is the set consisting of triplets of  $P_{Expected}[A_i^{-1}]$ ,  $D_{Expected}[A_i^{-1}]$ ,  $E_{Expected}[A_i^{-1}]$  of the  $N$  instances of  $X$  in the database, we can say that it is the set of prosodic expectations for the Optimal Unit of  $X$  by the units which are most likely to

precede  $X$  in an given random text. So, the Cummulative or the Global Prosodic Mismatch Function (GPMF) is calculated as shown below

$$GPMF(X) = \sum_{i=1}^N \left\{ \left| 1 - \frac{P_x}{P_{Expected}(A_i^{-1})} \right| + \left| 1 - \frac{D_x}{D_{Expected}(A_i^{-1})} \right| + \left| 1 - \frac{E_x}{E_{Expected}(A_i^{-1})} \right| \right\} \quad \text{Equation. ( 5 )}$$

substituting (4) in (5), we get

$$GPMF(X) = \sum_{i=1}^N \left\{ \left| 1 - \frac{P_x}{P(A_i)} \right| + \left| 1 - \frac{D_x}{D(A_i)} \right| + \left| 1 - \frac{E_x}{E(A_i)} \right| \right\} \quad \text{Equation. ( 6 )}$$

Ideally the Optimal Unit should have prosodic features  $\langle P_x, D_x, E_x \rangle$  such that  $GPMF(X)$  is zero. But it is practically impossible due to the absolute addition unless all the  $N$  occurrences of  $X$  have equal pitch, duration and energy. So, by definition, the optimal unit is selected to be the one that minimizes GPMF.

The above selection criteria is referred to as Algorithm F. The database consisting of Optimal Unit for each basic unit is referred to as Database F.

We must mention here that, by using Optimal Units, we are bettering the outcome in the most probable cases. However we are also worsening the outcome in the rare cases. Validity of this approach is arguable. We have conducted perceptual tests to establish this validity.

## 2.2. Perceptual Tests on Databases A, E and F

Perceptual tests were conducted to compare databases A, E and F. We synthesized 10 sentences of around 10 words per sentence from each of these 3 databases. 8 test subjects were asked to score each of the 10 utterances from the 3 databases on a scale of 0 (Worst) to 5 (Best).

**Table 1: Perceptual Evaluation of Databases A, E and F**

S. No.	Database A		Database E		Database F	
	Avg.	Var.	Avg.	Var.	Avg.	Var.
1	1.875	0.411	1.875	0.696	3.000	0.857
2	3.000	0.857	3.125	0.696	2.875	0.411
3	1.750	0.500	2.500	0.857	3.000	0.571
4	1.750	0.214	3.125	0.125	2.250	0.500
5	2.250	0.214	2.500	0.571	3.250	0.500
6	1.875	0.696	2.125	0.696	2.750	0.786
7	3.750	0.786	4.375	1.125	4.125	0.411
8	1.750	0.214	2.000	0	2.375	0.268
9	2.000	0.571	2.125	0.696	2.125	0.696
10	2.500	0.286	3.625	1.125	3.250	0.500
<b>Avg.</b>	<b>2.250</b>	<b>0.475</b>	<b>2.738</b>	<b>0.659</b>	<b>2.900</b>	<b>0.550</b>

The results of the test conducted are summarized in Table 1. The score (averaged across all 8 subject) and the variance of scores for each sentence for all the three synthesizers are listed.

We can clearly observe that Database F is the best performing database. Further the score of Database F are more consistent than that of Database E. Database E required manual work in marking the proper unit boundaries whereas Database F is generated automatically. Considering all these observations, we have decided to use Algorithm F for our further experiments with Unit Pruning.

### 2.3. Scaled down Synthesizer based on Database F

Using the database F created above, a scaled down synthesizer was developed for a handheld device. The size of the Database F for a Hindi Voice is 1.27 MB and contains 2786 unique syllable and phone units of the type C, V, CV, VC, CVC, CCV and VCC where V is a vowel and C is consonant. The units are GSM coded at 8 KHz.

The text given for synthesis is broken up into syllables using the syllabification rules given in [8]. The required syllable or the nearest sequence of syllable and phones for each of the syllable in the text is picked up from database F and is decoded and concatenated in the sequence without prosodic modification. Hence as desired a LMDS was built.

## 3. Automatic Unit Pruning

Moving further with the aims that we set at the start of this paper, we attempt to prune off the units that do not contribute significantly to the required prosodic diversity of the database.

### 3.1. Ranking the Units for Pruning

A reduction in the size can be attained by pruning away some of the instances of the units by grading the units on a score of their desirability (or undesirability) in the database. Also we can remove units that are very similar to the other instances because they do not add anything new to the prosodic diversity. Based on these two requirements we have developed a measure to score each instance of a unit.

#### 3.1.1. Using GPMF to measure Undesirability

We have used Algorithm F to find the Best Unit in building the LMDS. The unit having the minimum GPMF is taken to be the best or the most desirable one. Hence GPMF can be used as measure of undesirability. For pruning we use a weighted GPMF with different weights for each of the prosodic features. For each instance X of unit A having N instances, we calculate the Undesirability ( $U_x$ ) as

$$U_x = \sum_{i=1}^N \left\{ \begin{array}{l} W_{PITCH} \times \left| 1 - \frac{P_x}{P(A_i)} \right| + \\ W_{DURATION} \times \left| 1 - \frac{D_x}{D(A_i)} \right| + \\ W_{ENERGY} \times \left| 1 - \frac{E_x}{E(A_i)} \right| \end{array} \right\} \quad \text{Equation ( 7 )}$$

Typical values of weights that we have used are  $W_{PITCH} = 2.5$ ,  $W_{DURATION} = 2.5$ ,  $W_{ENERGY} = 1.0$

#### 3.1.2. Repulsion between units due to prosodic similarity

$$R_x = \sum_{j=1}^M \left\{ \begin{array}{l} W_{PITCH} \times \left| 1 - \frac{P_x}{P(A_j)} \right| + \\ W_{DURATION} \times \left| 1 - \frac{D_x}{D(A_j)} \right| + \\ W_{ENERGY} \times \left| 1 - \frac{E_x}{E(A_j)} \right| \end{array} \right\} \quad \text{Equation ( 8 )}$$

To prune units that are very similar to each other from the database, we define a repulsion term in our unit scoring

expression. We hypothesize that all instances which get selected to the database repel all the other units and that the repulsion between two units is proportional to the prosodic mismatch between the two units. So total Repulsion ( $R_x$ ) on the instance X of unit A due to M units of A already selected into the pruned database is given in Equation (8).

The higher the repulsion term for each instance the more it is desirable to have the unit in the database. So  $U_x$  and  $R_x$  are opposing forces in the scoring of an instance. The combined measure to score that we use is

$$S_x = U_x - (W_{REPULSION} \times R_x) \quad \text{Equation ( 9 )}$$

where  $W_{REPULSION}$  is the weight given to the repulsion term.

### 3.2. Creating the Pruned Database

We prune the database by keeping some percentage of the top most desirable instances of each unit. Some of the parameters that we use to specify the units that are to be kept are

*Phone/Syllable Usage Percentage:* Percentage of Instances of each unit to be kept in the database

*Minimum Phone/Syllable Count:* Minimum number of Instances of each unit to be kept in the pruned database. If the number of instances is less than this, then all instances are kept in the pruned database.

*Maximum Phone/Syllable Count:* Maximum number of Instances of each unit to be kept in the pruned database.

Using these parameters, the number of instances to be kept in the pruned database for each of the available unit in the database is calculated. The required instances are picked up by iteratively scoring each of the unselected instances of the unit using  $S_x$  as in Equation (9). At each iteration, the instance with minimum score is added to the database until the required number of instances is collected.

### 3.3. Heuristic for Obtaining Database of a Desirable Size

Table 2. Database Description Table

DB Name	Syl %	Phone %	W <sub>REPULSION</sub>	Size in bytes	Instance Count
DB1	90	10	0.5	34311940	22807
DB2	70	10	0.5	31171806	20777
DB3	50	10	0.5	28123724	18741
DB4	30	10	0.5	24877610	16552
DB5	10	10	0.5	22282862	14786
DB6	10	10	2.0	22343714	14786
DB7	30	10	2.0	24940186	16552
DB8	50	10	2.0	28225282	18741
DB9	70	10	2.0	31363732	20777
DB10	90	10	2.0	34486744	22807
DB11	30	5	0.5	23856974	14861
DB12	30	15	0.5	25593444	17680
DB13	30	20	0.5	26174070	18528
DB14	30	25	0.5	26694976	19253
DB15	30	30	0.5	27115152	19791
DB16	30	30	2.0	27177728	19791
DB17	30	25	2.0	26757552	19253
DB18	30	20	2.0	26236646	18523
DB19	30	15	2.0	25656020	17680
DB20	30	5	2.0	23919550	14861

The sizes for several pruned databases derived from the Hindi Speech Corpus created using different value of the mentioned parameters are shown in Table 2 below. Also the

total number of instances in each of the databases is mentioned. The parameters being modified are Phone/Syllables usage percentage and  $W_{REPLUSION}$ . Maximum Phone/Syllable Count is set to 100, Minimum Phone Count is 1 and Minimum Syllable Count is 10.

We can observe that  $W_{REPLUSION}$  does not have significant effect on the size of the database. The minimum size of database available here is DB5. The Maximum Size is that of DB1. The effects of Phone / Syllable usage percentage between these sizes can fairly be estimated to be linear. The empirical formula for the database size based on the Phone and Syllable usage percentages can be taken to be:

$$Size = 22282862 + (P - 10) \times 108606 + (S - 10) \times 150363$$

- Equation (10)

Here P and S are Phone and Syllable usage percentages respectively. Using this equation, we have found that the maximum error in approximation of size of above databases is less than 3.5%.

#### 4. Obtaining an Optimal Database

Table 3. Perceptual Evaluation of Various Databases

Sentence No ↓	Syllable Usage Percentahe				
	10%	30%	50%	70%	90%
<b>For <math>W_{REPLUSION} = 0.5</math></b>					
DB Name >	DB5	DB4	DB3	DB2	DB1
1	2.875	2.875	3.000	2.250	1.750
2	3.250	3.250	3.250	2.875	2.500
3	4.125	4.125	4.000	4.000	3.875
4	2.375	2.375	2.500	1.875	2.375
5	3.000	3.000	2.625	2.500	2.375
Average >	<b>3.125</b>	<b>3.125</b>	<b>3.075</b>	<b>2.700</b>	<b>2.575</b>
<b>For <math>W_{REPLUSION} = 2.0</math></b>					
DB Name >	DB6	DB7	DB8	DB9	DB10
1	2.875	2.750	3.000	2.875	3.250
2	3.500	3.750	3.875	3.750	3.125
3	3.625	3.625	3.625	3.375	4.125
4	2.500	2.875	2.625	3.000	2.625
5	3.625	3.625	3.750	3.500	3.500
Average >	<b>3.225</b>	<b>3.325</b>	<b>3.375</b>	<b>3.300</b>	<b>3.325</b>

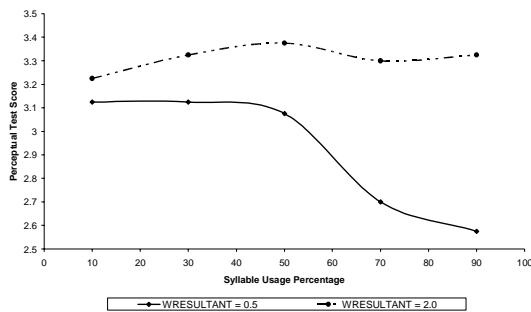


Figure 1. Perceptual Test Scores vs. Syllable Usage

We have conducted perceptual tests on the database DB1 to DB10. 8 native Hindi speakers ranked 5 sentences of about 8-10 words synthesized from each of the 10 databases on a scale of 0 (worst) to 5 (best). Lab conditions were used for the testing. Databases DB1 to DB5 have  $W_{REPLUSION} = 0.5$  and varying Syllable Usage Percentage. Databases DB6 to DB10

have  $W_{REPLUSION} = 2.0$ . The results of the tests are summarized in Table 3. The indicated scores are the average of the scores given to each sentence by the 8 subjects. Also the plots of Average Score of each database against the Syllable Usage Percentage are shown in figure 1.

We can observe that higher values of  $W_{REPLUSION}$  seem to be better which is justified as it leads higher diversity of prosodic features in the database. Further a database created with syllable pruning of around 45% should have the best perceptual performance. Also, reduction of syllable usage to 30% should be optimal in terms of both size and quality.

#### 5. Conclusion

We have proposed development of a tiny synthesizer as a by-product of a Unit Selection Speech Synthesizer. Various algorithms for selection the best instance of a unit are proposed and evaluated. Algorithm F is found to be the best of the proposed approaches. Also pruning of Unit Selection databases to be able to deliver a database of any required size is produced. Parameters for an optimal database are suggested. We have also observed that pruning not only reduces the size of the database but also improves the quality of synthesis to some extent by removing dirty units.

Further we must explore the effect of various distance measures on pruning of databases. Also LMDS needs to be incorporated with intonation and duration models to produce a very natural sounding speech from a tiny synthesizer.

#### Acknowledgement

We want to thank Prof. Rajeev Sangal for his inputs and discussions from time to time. Also we thank the Volunteers for perceptual tests for their contribution.

#### 6. References

- [1] AT & T Natural Voices™ Text to Speech System, <http://www.naturalvoices.att.com/>
- [2] Kenney Ng, "Survey of data-driven approaches to Speech Synthesis", *MIT Area Paper, October, 1998*
- [3] Alistair Conkie, "A robust unit selection system for speech synthesis", *In Joint Meeting of ASA/EAA/DAGA, Berlin, Germany, March 1999*
- [4] Alan W. Black and Nick Campbell, "Optimizing selection of Units from Speech Databases for Concatenative Synthesis", *In Proceedings of Eurospeech 95, vol 1., pp. 581 - 584, Madrid, Spain, 1995*
- [5] Alan W. Black and Paul A. Taylor, "Automatically Clustering Similar Units for Unit Selection in Speech Synthesis", *in Proceedings of Eurospeech 97, vol. 2, pp. 601-604, Rhodes, Greece, 1997*
- [6] Sanghun H. Kim, Youngjik J. Lee, and Keikichi Hirose, "Pruning of Redundant Synthesis Instances Based on Weighted Vector Quantization," *in Proceedings of Eurospeech 2001, vol. 3, pp. 2231-2234, Aalborg, 2001*
- [7] Yong Zhao, Min Chu, Hu Peng and Eric Chang, "Custom-Tailoring TTS Voice Font-Keeping Naturalness when Reducing Database Size", *in Proceedings of Eurospeech 2003, pp. 2957-2960, Geneva, 2003*
- [8] S P Kishore, Rohit Kumar and Rajeev Sangal, "A Data Driven Synthesis Approach For Indian Languages using Syllables as Basic Unit", *in Proceedings of Intl. Conf. on NLP (ICON) 2002, pp. 311-316, Mumbai, India, 2002*