

# Improving Preselection in Unit Selection Synthesis

*Alistair Conkie, Ann Syrdal, Yeon-Jun Kim, Mark Beutnagel*

AT&T Labs – Research  
Florham Park, NJ USA

{adc,syrdal,yjkim,mcbl}@research.att.com

## Abstract

Unit selection synthesis is a method of selecting and concatenating speech segments from a large single-speaker audio database to synthesize utterances. Selection is based on assigning target and concatenation costs to units and then finding a lowest cost sequence of units that will synthesize a given utterance. In order to synthesize efficiently, it is necessary to limit the number of units considered in the unit selection cost network, a part of the process called preselection. This paper examines the role of preselection in unit selection synthesis. We refine the existing process of preselection by adding multiple phone sets to the list of features considered. We present experimental results that demonstrate better database usage and significantly increased synthesis quality using this new method.

**Index Terms:** speech synthesis, unit selection, preselection costs

## 1. Introduction

Many modern speech synthesizers use concatenative methods to generate audible speech from text input. Currently the most popular version of concatenative synthesis, called unit selection [1], uses a large inventory of recorded speech in which multiple variants of units are available for concatenation. Some unit selection synthesizers e.g. [1] use phones as the minimal units for concatenation, but the introduction of half-phones as minimal units [2] allowed for joins either in the middle of a phone or at a boundary between phones and improved synthesis quality. For simplicity our work is described here in terms of phones, but the underlying synthesizer is implemented in terms of half-phones.

Typically in unit selection two cost functions are used in calculating an optimal set of units to form an utterance. The **target cost** measures how close (in terms of  $f_0$ , duration and other parameters) an individual database unit is to a synthesis specification. The **join cost** is a measure of the degree of perceived discontinuity between two units to be joined. The sequence of units with the best, i.e. lowest, overall cost (sum of target and join costs) is assumed to be the sequence of units that results in the best quality synthesis. This sequence of units is concatenated together and the audio file output. The more correlated the costs are to listener perception, the better the quality is likely to be.

A database can consist of many thousands or even millions of units. In order to have a synthesizer that can respond quickly to input text it is necessary to refine the basic unit selection concept of searching a network of costs to find the lowest cost path by introducing a **preselection** mechanism. The preselection method can be regarded as an adjunct to the target cost calculations. The preselection method assigns an approximate context-based cost to individual units prior to calculating the complete target cost. The costs are used for the purpose of

pruning the number of possible candidates, which may number several thousand for a particular phone type, down to a number which can be used efficiently in the network. We have previously considered methods whereby this preselection process can be optimized by precalculating likely candidates [3]. Using these methods it is only necessary to do preselection on a small number of units known to be relevant.

In [4] decision tree methods are used to achieve essentially the same goal of considering only a small number of relevant units at synthesis time.

In [5] we explored achieving closer agreement between the unit specification produced by the front end and the labeled speech database as well as making finer distinctions in the phone set by distinguishing between pre- and post-vocalic phones. This combination successfully improved synthesis quality. There is a small increase in complexity due to the fact that there is effectively a new phone set for the database necessitating relabeling. There is also the possibility that, due to preselection only being possible for an exact phone match, units that might be useful but don't have the correct pre- or post-vocalic designation may be eliminated from consideration.

The work described here refines the preselection by introducing extra phone sets. There may be one or several. The new phone sets are introduced as additional features in the speech database index. The preselection process itself is minimally modified to benefit from the new features. The net result is to provide finer control over the selection of units in terms of symbolic rather than direct acoustic measures. The mechanism described is very general.

By permitting finer distinctions between unit types, without eliminating units that are not an exact match, we leverage the audio in the database more effectively for synthesis, hence increasing output speech quality.

The remainder of the paper will describe the preselection process in more detail, indicating where changes were made relative to the previous approach. This is followed by a description of the experiments that were carried out to test the effectiveness of the modifications.

## 2. Target Costs and Preselection

The target cost is intended to be a measure of the suitability of a particular unit for use in synthesis. Input text is converted in the front end to an acoustic and symbolic specification in terms of phone identity, duration and  $f_0$ , and optionally including other potential feature quantities such as energy or allophone type.

Typically there is a weight training process (based on acoustics) [1] whereby an attempt is made to relate these specification features to perceptual differences. Using the trained weights and the features considered relevant to unit selection we can estimate the target cost for any database unit for any

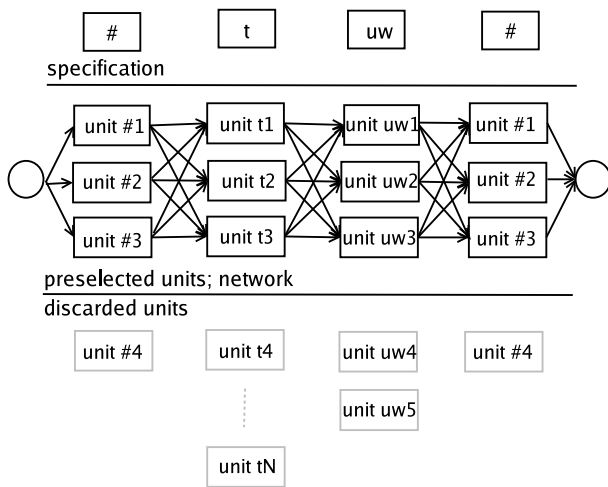


Figure 1: *Preselection and search process.*

synthesis context/specification. In practice, rather than perceptual differences being used, cepstral distance measures are substituted as an approximation.

Figure 1 illustrates the important aspects. Once a specification is known, lists of matching units in the database (without regard to context) are retrieved. The preselection cost is calculated for each unit. The lowest cost  $n$  units (for the example  $n$  is 3) are retained, and the remaining units no longer considered. The full target and join cost calculations are done only for the preselected units, and finally the lowest cost path is calculated.

The preselection step is added into the mix to reduce the number of candidate units for unit selection. The number of join costs to be calculated for each unit is  $O(n^2)$  (where  $n$  is the maximum number of candidate units considered in the Viterbi network) so preselection is a very necessary step. The preselection step for a particular unit is  $O(N \log N)$  where  $N$  is the number of phones of that type in the database. This can be optimized as described in [3]. Determining join costs is the most expensive part of the calculation.

Preselection works on the basis of broad phonemic contexts rather than acoustic information. Broadly speaking a context of  $\pm 2$  phonemes is considered as the range of effectiveness in determining modifications to the form of a phone. A calculation is done where the desired sequence of units and the database sequences of units are compared. The nearer phones are weighted most heavily, the more distant less heavily. Matching the target phone is so important that we do not consider other options than a direct match.

### 2.1. New features

Since the preselection process uses broad phonemic categories and the database is large there is a concern that adequate preselected units may be found near the beginning of the database and that units later in the database are never accessed. Consequently we may not be using the full potential of the database. With this in mind we modified preselection to be more discriminating, with the intention of using more of the database and increasing synthesis quality.

A unit selection database in its most basic form consists of a large number of audio files and an index into the units within each file. The index can be represented in the form of a table of unit numbers, phonemes and other features.

Instead of dividing phonemes into hard categories as in [5] we chose to use various specialized phone sets, incorporated into the unit selection index in the form of extra features.

We begin by adding four extra phone set features. One of them is the modification, described in [5], recast as a feature. These features take the form of variant phone set, expressing more detail than the standard phone sets. In this way allophonic variations can be considered as part of preselection.

Note that there is some commonality between features; they are not required to be independent.

It is important to note also that the front end was modified to produce the new features.

The features are listed below.

### 2.2. Word boundaries

We add a feature where word boundary positions are associated with phonemes. So, “**the cat**” would be represented as “[**dh ax** | **k ae t**]” (rather than “**dh ax k ae t**”), and “[**ay**]” would be a representation of the word “**T**” (compared with “**ay**”).

### 2.3. Initial consonant clusters and glottals/flaps

For this feature we co-opt an aspect of the Festival speech synthesis system [6]. There are two parts. For the first part we distinguish between initial consonant clusters and other consonant clusters. Some examples:

“**string**” -> “**s\_ .t\_ .r ih ng**”, but “**last**” is represented as “**l ae s t**”; “**prime**” -> “**p\_ .r ay m**”

Additionally, at word boundaries where a vowel is adjacent to a stop a \$ is added to the stop: eg “**eat it**” would be “**iy t\$ ih t**”. The underlying assumption is that these diacritics, based on initial consonant clusters being distinct and the possible occurrence of glottal stop or flap allophones of **t** as in example “**iy dx ih t**” are useful for the Festival synthesis and so could be useful in a unit selection context.

The diacritics can be combined so one might see, for example, **\$t\_** as a possible “decorated” phoneme feature. This would occur where the **t** is part of a word-initial consonant cluster that follows a word ending with a vowel.

### 2.4. Content/Function

We distinguish between phonemes from content and function words. We want to avoid phonemes from function words being used in content words, particularly in stressed positions. There is evidence from speech recognition that this distinction is advantageous [7]. If a word is considered to be a function word the phonemes are labeled with an additional **\_f** in the “**func**” feature. So “**m\_f**” would be the function word version of “**m**”. And “**the**” would become “**dh\_f ax\_f**”.

### 2.5. Enhanced Phones

Lastly we make the enhanced phones described in [5] into a feature and use ARPAbet phonemes for the basic unit phone categories. This enhanced phone set distinguishes pre- and post-vocalic consonants. The syllabification scheme adopted will influence where the feature is applied and must be consistent for best results. As an example of usage, “**last**” would be transcribed “**l ae s- t-**”, whereas “**star**” would be transcribed “**s t aa r-**”.

## 2.6. Algorithm modification

The preselection process is modified so that feature comparisons are possible based on the new phone features, and not exclusively on the standard phone set. Thus far experimentation on the appropriate contribution that new features should make to the weights has been limited.

The preselection cost has a component for context (and an implicit component for phoneme identity). To this we add costs associated with the various specialized sub-types for the phoneme, as defined by the four new features. For the purposes of the experiment described here we adopted a simple difference penalty approach for the new features. When a requested feature is in disagreement with the corresponding database feature the cost is higher.

## 3. Database order

One of the concerns we examine in this paper is the relationship between preselection and full use of the speech database. Because preselection prunes the number of candidates for full unit selection we find that part of the database is effectively inaccessible. The inaccessible portions are common phonemes in common contexts, so their absence may not be important. However, given the importance of continuity in achieving high quality unit selection, synthesis may be functioning at less than 100% efficiency. We set out to make some measurements with different database orderings to compare with the feature refinements described above.

## 4. Experiments

### 4.1. Listening Test

A web-based subjective rating test was conducted. The same 20 test sentences were synthesized by each TTS system. Each sentence was rated for overall quality on a 5-point scale: 1(bad) to 5 (excellent).

Three TTS versions were evaluated (all using same AT&T Natural Voices 4.1 front end):

- **Old baseline:** Reference voice as of 2006, female speaker of American English, using enhanced phones as described in [5]
- **New baseline:** A more recent version, with a different database composition (less than 10% different)
- **New:** Version using the new preselection scheme (same database composition as **new baseline**)

60 test sentences (20\*3) were rated by each listener. 29 AT&T employees volunteered as listeners. A total of 1740 ratings (60\*29) were collected during the test.

### 4.2. Database order

We built four voice databases. Two were built using our "standard" database order and two were built with the order reversed. This provides the maximum contrast between the two databases in terms of preselection. No other aspect of the system was modified. For each pair, one voice database carried the new features while the other was a reference or standard database.

A large body of text, corresponding to approx. 1.1 million phonemes was input to four different versions of unit selection and the resulting unit lists were collected and examined.

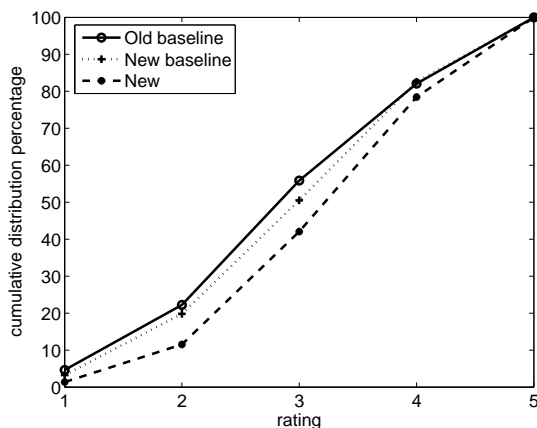


Figure 2: Distribution of ratings

## 5. Results

### 5.1. Listening test

Table 1: Listening test results

TTS	MOS	S.E.
New	3.67	.111
New baseline	3.44	.110
Old baseline	3.35	.122

Repeated measures ANOVA results found significant differences among TTS versions ( $F(2,56)=22.204$ ,  $p<0.0001$ ). Posthoc pairwise comparisons of TTS versions' ratings indicate that:

- **New** was significantly higher ( $p<0.0001$ ) than either baseline version.
- **New baseline** was significantly higher ( $p<0.039$ ) than **Old baseline**.

There were also significant differences in ratings among sentences ( $F(19,532)=15.459$ ,  $p<0.0001$ ) and a significant TTS by sentence interaction ( $F(38,1064)=7.902$ ,  $p<0.0001$ ).

An alternative way to view the data is in terms of cumulative counts by rating as shown in Figure 2. We see for example that almost 60% of the ratings for Old baseline were 3 (fair) or lower, whereas 40% for New TTS were rated 3 or lower, while 60% were rated 4 or 5 (good or excellent). If we consider the ratio of good and excellent ratings (4 & 5) to poor and bad ratings (1 & 2) in Table 2 we see that there is a striking difference, with the new voice having a much higher ratio than the two baseline versions.

Table 2: Ratio of good and excellent to poor and bad

TTS	ratio
Old baseline	2.0
New baseline	2.5
New	5.0

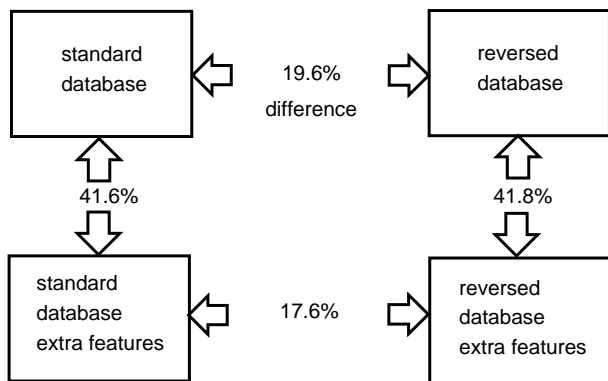


Figure 3: Differences from synthesizing with different databases.

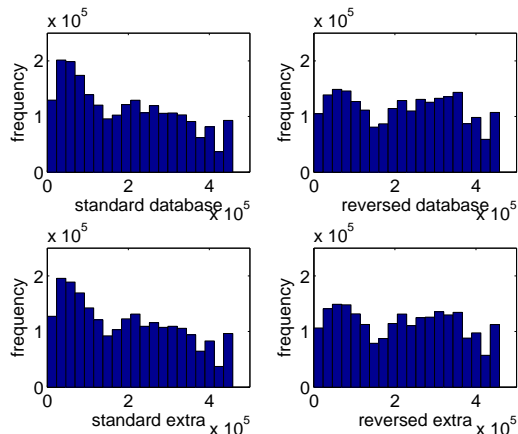


Figure 4: Frequency of use of database units, by index.

## 5.2. Database ordering results

In Figure 3, we compare database configurations by measuring the percentage of output units that change, for identical input, when different databases are used. Adding features has a more pronounced effect on the synthesis output (41.6% and 41.8%) than changing database order. Database order is less important (17.6% vs. 19.6%) for the database with the extra phone set features.

In Figure 4, the frequency of usage of database units is illustrated. The x-axis in each case shows position in the database. The left pair of histograms are from databases in the standard order, the right pair are from the databases in reversed order. The top two histograms are without the new features, the bottom two with the new features. The effect of database order (left pair vs. right pair) is much more visible than the effect of the extra features (top pair vs. bottom pair). Reordering results in units being chosen from different sections of the database, while adding extra features causes different units to be chosen but without any marked shift from one section of the database to another.

## 6. Discussion

Our new method offers clear improvements in perceived synthesis quality. Better preselection leads to higher quality synthesis.

The new method allows us to preselect units more effectively to better use more of the database.

The results suggest that by having broad phone categories and making distinctions within these categories on the basis of weights we have more flexible control over what units are good candidates for a particular context than if we were using, for example, decision tree based categories [4].

It is interesting to compare this work with [4]. Both methods attempt to select a small subset of units for synthesis. Black and Taylor use decision trees, with acoustic differences as a similarity measure, whereas our relies on finer context-based phone distinctions in conjunction with preselection optimization. Both try to improve on existing synthesis, and reduce the number of low-scoring utterances. The results from [4] are somewhat inconclusive, and our results compare favorably.

We also found that having multiple phone sets allows flexibility in the construction of the unit selection in general.

The new method is also language independent in the sense that we are free to add new features we deem appropriate for a particular language

## 7. Conclusion

This paper presented a refinement of our existing process of preselection by adding multiple phone sets to the list of features considered. Experimental results demonstrate better database usage and significantly increased synthesis quality using this new method.

## 8. Acknowledgement

The authors would like to thank Srinivas Bangalore for setting this work in motion.

## 9. References

- [1] A. Hunt and A. Black, "Unit selection in a concatenative speech synthesis system using large speech database," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pp. 373–376, 1996.
- [2] A. Conkie, "A robust unit selection system for speech synthesis," *Joint meeting of ASA, EAA, and DAGA*, p. 1PSCB\_10, 1999.
- [3] A. Conkie, M. Beutnagel, A. Syrdal, and P. Brown, "Preselection of candidate units in a unit selection-based text-to-speech synthesis system," *International Conference on Spoken Language Processing ICSLP 2000*, vol. 3, pp. 314–317, 2000.
- [4] A. Black and P. Taylor, "Automatically clustering similar units for unit selection in speech synthesis," *Proc. EUROSPEECH*, vol. 2, pp. 601 – 604, 1997.
- [5] Y.-J. Kim, A. Syrdal, A. Conkie, and M. Beutnagel, "Phonetically Enriched Labeling in Unit Selection TTS Synthesis," *International Conference on Spoken Language INTERSPEECH-2006*, pp. paper 2055–Tue3BuP.6, 2006.
- [6] A. Black and P. Taylor, "The Festival Speech Synthesis System: system documentation," *Technical Report HCHC/TR-83*, 1997.
- [7] K.-F. Lee, *Automatic Speech Recognition: The Development of the Sphinx Recognition System*. Kluwer, 1990.