# A PHONOLOGICALLY MOTIVATED METHOD OF SELECTING NON-UNIFORM UNITS

*A. P. Breen & P. Jackson*

BT Laboratories, Martlesham Heath, Ipswich, IP5 3RE, U.K.
Email: {andy.breen, peter.jackson}@bt-sys.bt.co.uk

## ABSTRACT

This paper describes a method for selecting units from a database of recorded speech, for use in a concatenative speech synthesiser. The simplest approach is to store one example of every possible unit. A more powerful method is to have multiple examples of each unit.

The challenge for such a method is to provide an efficient means of selecting units from a practical inventory, to give the best approximation to the desired sequence in some clearly specified way. The method used in BT's Laureate system uses mixed N-phone units. In theory such units could be of arbitrary size, but in practice they are constrained to a maximum of three phones. It dynamically generates the unit sequence based on a global cost. Units are selected using purely phonologically motivated criteria, without reference to acoustic features, either desired or available within the inventory.

## 1. INTRODUCTION

### 1.1 Background

Concatenative synthesis systems generate speech from a unit inventory of sounds. In recent years, this has become a favoured method of synthesis, due to developments in speech modification algorithms and the continuing downward trend in the costs of computer processing capability and memory.

The process of unit selection for such a system is predicated upon the fact that there exists a finite inventory of sounds with an adequate coverage of the phonemic constituents of the desired language and accent. What is considered to be adequate varies considerably among researchers, as it depends to a large extent on the criteria used in the unit selection processes and the design constraints imposed by the synthesis system.

The most common approach is to define a set of synthesis units (e.g. diphones, triphones and demi-syllables [4][5]) and from this definition create a speech database that contains at least one example for each unit. Such an approach goes some way to overcoming some of the well known problems of coarticulation, where the production of one phone can be highly influenced by its preceding and following neighbours, and other contextual effects [1].

### 1.2 Design Criteria

The primary goal of a unit selection process is to make best use of the available information contained within the finite data set. However, other practical factors such as processing time and the desire to minimise the number of unit discontinuities also play an important part in the design of a system. For the method described in this paper, the underlying design criteria are as follows:

- The selection process must make maximum use of the available data. The adequacy of the database may vary, but the minimum selection criterion is that the database should contain at least one example of every phoneme that can be selected.
- The selection process must be efficient. Efficiency is used here to mean that the process must be completed in a realistic time. Clearly what is considered as a realistic time will depend on the envisaged applications.
- The selection process will use global selection criteria to minimise the number of unit concatenation discontinuities observed within a specified body of speech.
- The selection process should exhibit graceful degradation, such that if an exact match to the desired unit environment cannot be found, the best compromise choice will be made.

### 1.3 Method Overview

The unit selection process used by the Laureate system [2] is carried out in two stages. The first stage consists of a database search, where the best possible unit candidates are identified by means of a distance metric. The database search technique is described in section 3 of this paper. The method of calculating the distance metric, which is based on purely linguistic criteria, is described in the companion paper [3].

The second stage uses a method of dynamic programming to determine the optimal set of units based on information provided by the database search and on signal processing constraints. This is described in section 4.

First however, the approach to the database design will be examined.

## 2. DATABASE DESIGN

### 2.1 Database content

An alternative to the fixed inventory approach previously mentioned, is to dynamically select segments of speech from a very large corpus. Implicit in this approach is the need to ensure that the corpus has sufficient coverage. Between these two approaches lie a number of methods that attempt to optimise the design of the database based on specific linguistic criteria. Inevitably, the nature of the recorded database will depend upon

the underlying assumptions of the unit selection process. An additional consideration for many researchers is the size of the database. Practical limitations on size significantly affect the type of database recorded and hence the sophistication of the unit selection process employed.

The speech database design used in Laureate has similarities with the approaches mentioned above. Large speech databases are considered difficult to maintain and even more difficult to annotate reliably. As a result a fairly simple adequacy criterion is employed. The speech database is designed such that it contains at least one instance of every diphone permitted by the pronunciation model. However, it differs from many such databases in that it is not composed from diphones embedded within a set of carrier phases. Instead, the database consists of phonetically rich passages. Diphone coverage only represents a minimum adequacy criterion. The unit selection process is not restricted to selecting diphones but is free to select N-phone units.

## 2.2 Speaking style

Comparatively little research has been conducted into speaking styles within the synthesis process, yet inventory based systems are particularly sensitive to the manner of speech recorded for the database. Investigations of synthesis style have centred predominantly in the area of emotional synthesis [7], where researchers have concentrated on simulating basic emotions such as fear, happiness, sadness and boredom.

In designing the speech database, care has been taken to ensure that the recorded speech does not exhibit any strong speaking style, but maintains a neutral, placid quality. Where feasible and appropriate, specific speaking style effects may be imposed on the speech as part of the post selection synthesis process.

Storing different styles of speech in the unit database is not in itself a problem. The real difficulties arise when differing styles of speech are under-specified. As a result, the selection processes may choose segments of speech from widely differing styles. When synthesised, the patchwork nature of such concatenated speech is clearly audible. Storing clearly specified speaking styles is only useful if the synthesiser is capable of using style as a selection criterion, and the database has adequate coverage of that style.

Speaking style represents an extreme example of the problems encountered in the appropriate specification of speech data for synthesis. The choice of features used to specify a particular database and the annotation strategy employed in labelling the database also significantly affect the performance of the unit selection process.

# 3. DATABASE SEARCH

## 3.1 General Description

This stage of the selection process examines the entire inventory of sounds and identifies all possible candidate units. In reality, the speech data plays no part in the selection process - rather it is the time-aligned annotations associated with the speech which are used. The richness inherent in the speech database is therefore completely dependent on the sophistication and accuracy of the information associated with each annotation. Within this framework, two sounds are considered identical if they have identical annotations. The annotation information is a combination of the nominal phonetic attributes of a sound and the phonological environment in which it sits.

For implementation efficiency during speech synthesis, the candidate search is not performed directly on the raw annotation data, but on a pre-processed structure - the *phoneme context tree*. The remainder of this section will describe the manner in which the context tree is constructed and the search technique.

## 3.2 Building the phoneme context tree

A unit, as understood by the context tree, is simply a sequence of phones. The maximum length of this sequence is determined by the size of the *context tree window*. The context tree window defines the number of phones examined during a database search. As an example, a window size of five would contain five phones, a central phone bounded by two neighbours on either side. A context window must be symmetric, but the maximum width of a context window is purely a matter of computational convenience.

When constructing a context tree, each phone in the annotation database is examined in turn, embedded within a window of the defined size. Once built, the phoneme context tree has a structure similar to that shown in Figure 1.

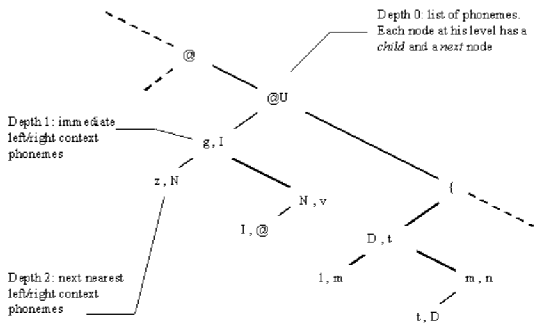


**Figure 1.** Schematic representation of a portion of a phoneme context tree for a window size of 5.

Each node on the first level (depth 0) of the tree consists of a simple index to a particular phoneme. If the synthesis system contained forty-two basic phonemes, then there would be forty-two nodes at the first level, one for each phoneme symbol. Nodes at this depth are not placed in any specific order. Each node at depth 0 must have at least one child node (depth 1 of the tree). The nodes at depth 1 of the context tree record all combinations of immediate left / right neighbours for the centre phoneme indexed by the parent node at depth 0 in the tree. The contents of each node at depth 1 are shown in Table 1.

| | |
|---|---|
| **Left Context** | Feature vector containing information on immediate left hand phone (wrt parent) |
| **Right context** | Feature vector containing information on immediate right hand phone (wrt parent) |
| **Next** | Link to next (current depth) neighbour pair |
| **Child** | Link to (current depth +1) nearest neighbour pair |

**Table 1:** Information stored in nodes at level 1 of the context tree.

The left and right context feature vectors contain sufficient phonetic and phonological information to uniquely specify the environment and the phonetic element embedded in that environment. Intermediate levels of the tree (if they exist) below the first level and excluding the last level contain nodes with the same structure as that shown in Table 1.

Nodes at the lowest level of the tree are leaf nodes and as such contain significantly more information than the proceeding levels. Table 2 and Table 3 show the information contained in each leaf node.

| | |
|---|---|
| **Left context** | Feature vector containing information on immediate left hand phone (wrt parent) |
| **Right Context** | Feature vector containing information on immediate right hand phone (wrt parent) |
| **Next** | Link to next (current depth) neighbour pair |
| **Data array** | M-element array *(See Table 3).* |

**Table 2**: Information stored in nodes at the lowest level of the context tree (leaf nodes). M = window size.

| | |
|---|---|
| **Phoneme index** | Index to specific phoneme in symbol set |
| **Phone duration** | Duration of phone |
| **Offset within data file** | Position of phone in data file |
| **Data file label** | Name of data file containing specified phone |

**Table 3:** Information stored in each element of the M-element array in nodes at the lowest level of the context tree.

During construction, the tree structure is first initialised with depth 0 nodes. Once initialised, each of the windowed blocks of text is placed in turn in the tree. Information is progressively added to the context tree as the context window moves through the available annotation data. The build process is considered successful if, after all the annotation data has been presented to the tree, all depth 0 nodes have at least one active child.

The context tree is built once for each speech database, as part of an off-line data pre-processing stage. During speech synthesis however, the tree is repeatedly searched as part of the unit selection process.

## 3.3 Selection of candidate units

As previously stated, the concept of a unit within the phoneme context tree differs from the standard definition of a synthesis unit. The maximum unit size simply specifies the maximum sequence length of phonemes that can be drawn from a given context tree window. In fact, the search strategy makes no explicit use of the concept of a unit. During synthesis exhaustive searches are performed on the tree with differing bias conditions. It is these bias conditions that lead to the selection of specific unit types.

Consider the symbol string ABCBA. The centre symbol is C and the set of all possible candidate units is /BC_/, /_CB/, /BCB/ and /_C_/. The underscore is used to denote the positional bias of the unit within the context window. These units are termed left diphone, right diphone, triphone and centre phone respectively. The context tree search locates all, or a subset, of these unit types for each context window, depending upon the coverage of the data recorded within the tree. Candidate units are selected for each context window and placed into a workspace. An example of this is shown in Figure 2. Note that the phone unit is considered a degenerate case of the triphone unit and is only placed in the workspace when no suitable triphone is found.

| Centre Phoneme | # | t | Q | m | @ | s | # |
|---|---|---|---|---|---|---|---|
| Left Diphone | | #t_ | tQ_ | | m@_ | @s_ | s#_ |
| Right Diphone | _#t | _tQ | _Qm | | _@s | _s# | |
| Phone | _t_ | | | _m_ | _@_ | _s_ | _#_ |
| Triphone | | #tQ | tQm | | | | |

**Figure 2:** The workspace structure, populated with candidate units.

The cost function used in the above process is designed to select the unit with maximal match to the target context at each window in the workspace. The criteria used in this function include nominal features of the phonemes within the unit and its neighbours, as well as the features associated with the environment in which the unit lies. The latter class includes information such as stress, the presence of immediate word and syllable boundaries and position within the syllable (e.g. onset). The cost function is described in more detail in [3].

# 4. SELECTION FROM THE CANDIDATE UNITS

This stage of the selection process identifies the best choice of units from the candidates identified in the previous stage. Dynamic programming [6] is used to determine the best possible path through the workspace, starting from the last window in the input string. There are two ways that units can be joined - either overlapped or abutted. Units can be overlapped when the right context element of the first unit is the same as the left context element of the second unit – e.g. <ABC><CB> can be overlapped. Units are abutted when the right context element of

the first unit is not the same as the left context element of the second unit – e.g. <_CB><_C_> or <_C_><A#>.

A cost is calculated for every join. The cost for abutting units is higher than that for overlapping units. Therefore, the DP procedure will generally favour overlapping sequences of longer units, although the exact behaviour depends upon how well populated the workspace is with candidate units, and upon the type of cost function applied at each decision point. The process is illustrated in Figure 3.

| Centre Phoneme | # | t | Q | m | @ | s | # |
|---|---|---|---|---|---|---|---|
| Left Diphone | | #t_ | tQ_ | | m@_ | @s_ | s#_ |
| Right Diphone | _#t | _tQ | _Qm | | _@s | _s# | |
| Phone | _t_ | | | _m_ | _@_ | _s_ | _#_ |
| Triphone | | #tQ | tQm | | | | |

**Figure 3:** Path selection through the workspace structure.

The output of the unit selection process is a sequence of references to units. These may be used to extract segments from the database for use in the speech generation stage.

## 5. DISCUSSION

Many current synthesisers employ concatenation methods. This paper has presented a technique for the selection of units from a speech database, for use in such a synthesiser.

An important point about this method is that the choice of units is based on phonological, not acoustic features. There are a number of reasons for this. One is the requirement for efficiency defined in our design criteria. Extracting and processing actual acoustic features incurs far more computational cost than our approach. Furthermore, reliable extraction of acoustic features which are perceptually significant is also very problematic. Finally, there is an inevitable need for compromise when a match at the acoustic level conflicts with one at the phonological level – and this is very difficult to resolve without degrading the resultant generated speech.

The argument for the use of phonological features is based on the premise that the *context* of a speech segment, defined according to these features, will affect the acoustic realisation in a reasonably predictable and systematic way. Therefore, appropriate use of these features will aid optimum unit selection and hence the generation of fluent, natural sounding speech. In effect, this is a move towards comparing the intended linguistic function of the source and target segments. As an illustrative example of this, the pronunciation model used in the synthesiser does not differentiate allophonic variations of a phoneme, but relies on the environmental information to make appropriate selections. The richness and accuracy of the annotation data associated with the speech recordings as well as appropriate use of this information is critical in maximising the performance of such a method.

We have also touched upon one of the problems associated with this method - the issue of speaking style. The style currently used for speech databases is a compromise, for the reasons previously discussed and this is reflected in the style of the generated speech. The ability to generate a *variety* of speech styles presents one of the current challenges for researchers using unit inventory based systems.

## 6. REFERENCES


1. Gimson, A.C., *An Introduction to the Pronunciation of English*, Edward Arnold, 1989.
2. Page, J. H., Breen, A. P., "The Laureate text to speech system – architecture and applications", *Speech Technology for Telecommunications,* Chapman and Hall, 1998.
3. Breen, A. P., Jackson, P., "Non-uniform unit selection and the similarity metric within BT's Laureate TTS system". *Proc. Third International Workshop on Speech Synthesis, ESCA,* 1998.
4. Dutoit, T. et al, "The MBROLA project: towards a set of high quality speech synthesizers free of use for non commercial purposes", *ICSLP' 96*, 1996.
5. Jurgens, C., Wunderlich, M., "A comparison of different speech units for the German TTS system TUBSY", *Proc. Eurospeech '95,* 1995.
6. Rao, S.S., *Optimisation Theory and Applications*, Wiley Eastern Ltd, New Delhi, 1987.
7. Murray, I.R., Arnott, J. L., "Synthesising emotion in speech: is it time to get excited?", *Proc. ICSLP 96,* 1996