# Non-uniform Unit Selection through Search Strategy for Blizzard Challenge 2007

*Feng Ding, Jari Alhonen*

Nokia Research Center Beijing, Beijing, China
feng.f.ding@nokia.com

## Abstract

This present paper describes a non-uniform unit selection synthesis system for the Blizzard Challenge 2007. Non-uniform unit is used to maximize the length of unit sequence to be selected for the target sequence. In a minor modification from the previous implementation, a different search strategy is introduced to transfer a usual phoneme-based speech system to a non-uniform unit selection system, without big changes to the voice database. The front-end analysis results, such as syllable boundary, word boundary, and prosody phrase boundary, are utilized to search from different layers. The probable best small unit instance will be selected, gradually growing up to a longer unit. It is still possible to give up an original phoneme sequence existing in the database if that sequence mismatches the context significantly.

## 1. Introduction

Text-to-speech (TTS) technology can be applied whenever a computerized application needs to communicate with a human user. Typical examples of such applications include remote IVR, e-mail and SMS reading, audio books and gaming. The current speech synthesis efforts, both in research and in applications, are dominated by methods based on the concatenation of spoken units. New progress in the concatenative TTS technology is mainly made in from two directions, either by improving the synthesized speech quality in terms of intelligibility and naturalness, or by reducing the memory footprint and computational complexity to integrate the system into embedded system.

A lot of factors affect the quality of concatenative TTS systems. Unit selection [1] takes the benefit of a large inventory of recorded database. Multiple variants of acoustic units provide possibility for more complex context environment of acoustic units. It is always true that a bigger database will bring better quality. To utilize all kinds of variants efficiently, a clustering algorithm [2] groups the acoustic units according to their phonetic and prosodic context, offering efficient natural speech. Using Multisyn method [3], general purpose TTS system using the unit selection can be built.

As the basis of unit selection, the basic unit could be half phone, phone, diphone and syllable, etc. The characteristics of the language will affect the decision of basic unit. For example, Mandarin is a syllabic based language and it has regular CV structural syllables. Strong co-articulation can be found between phonemes in the same syllable, while co-articulations between phonemes across syllable boundaries are weaker. Thus tonal syllables can be chosen as the basic units in a Mandarin synthesis system. When choosing the basic unit, we should think about the whole system's footprint, and enough variants of the basic unit to cover different situations. The basic unit has a close link with the whole system's performance. The concatenation of units always causes quality degradation around the joint point, such as spectrum discontinuity, pitch change, etc. It is natural that we want to use as long as possible a unit sequence from voice database to minimize this kind of quality problems. In order to avoid the unnecessary concatenation, some non-uniform unit selection methods have already been investigated [4,5,6,7].The term non-uniform unit means a unit of variable length. During synthesis procedure, first the longer unit sequences are checked; if they are not provided, the system will roll back to use a shorter basic unit. What's the suitable length for a non-uniform unit also needs a lot of research.

In this paper, a new non-uniform unit selection method is implemented through a new search strategy for Blizzard Challenge 2007[8], which is an evaluation that compares the performance of different systems when trained on the same audio databases. The whole search procedure consists of searching in different layers. All units at different layers are taken as chunks. We used three layers: syllable, word and phrase. The basic idea is to get the best chunk candidates, then based on them move to the upper layer. The whole target sequence is finally decided at phrase level. The advantage of this method is that only the search strategy module needed to be revised, and the voice database could be left almost untouched.

The paper is organized as follows. Section 2 describes the overview of TTS system and synthesis procedure. Section 3 presents the new non-uniform unit selection method. Section 4 contains the voice building process with English speech database from ATR for Blizzard challenge 2007, followed by the conclusions.

## 2. Unit Selection TTS System overview

As shown in Figure 1, generally a concatenative TTS system includes a front-end module and a back-end module. The front-end module normalizes the text input and performs text analysis and prosody analysis to form a target sequence. The core question of concatenative TTS systems is how to get a sequence of units from a large voice database in which multiple variants of units are available for concatenation. After front-end analysis the input text will be transferred into a target phoneme sequence with or without other prosody information. It can be described as $T = \{t_1, t_2, ..., t_i, ...t_n\}$. In the database, one candidate sequence could be denoted as $U = \{u_1, u_2, ..., u_i, ...u_n\}$, where n denotes the total number of units to be synthesized. Two kinds of costs are defined to evaluate the distance between the candidate sequence and target sequence: target cost $C^t$ and join cost $C^j$. The target cost is an estimate of

the difference between a certain database unit and the target unit. It can be calculated as the weighted sum of phonetic and prosodic context. The join cost is an estimate of the quality of a join between units. The join cost mainly considers the acoustic characteristics of the two units. The total cost of using a candidate sequence can be presented as:

$$C = C^t + C^j$$

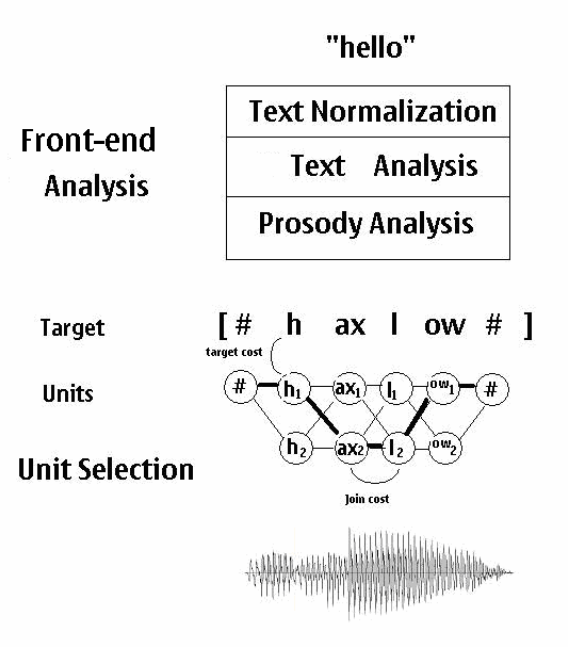


*Figure 1: unit selection based TTS system*

The unit selection is to find a suitable sequence $\hat{U}$, which satisfies $\hat{U} = \arg\min_{u} C$

Considering that $U$ is a sequence, the join cost can be:

$$C^j = \sum_{i=2}^{n} C_i^j(u_{i-1}, u_i)$$

where $C_i^j(u_{i-1}, u_i)$ is the join cost of adjoining units.

A set of acoustic features is selected to calculate the acoustic distance at the joined boundaries of two adjacent units. The feature set and join cost function can be explored through analyzing the correlation between the subjective perceptual listening results and the objective distance measure of certain feature sets. The feature set could include energy, pitch, and other features, such as line spectral frequencies or cepstral coefficients.

To find the suitable sequence, a search algorithm will be deployed to find the optimized result. Viterbi decoding is widely used in such a case.

## 3. Non-uniform unit selection through search strategy

### 3.1. The essentials of Non-uniform unit selection

Concatenative speech synthesis algorithms merely put together noncontiguous speech segments, wherefore the resulting prosody, which encompasses the three dimensions of pitch, duration, and energy, may not necessarily sound natural.

When two units are concatenated together, many typical problems occur:

- Speech intelligibility is greatly dependent on consonant-vowel transition parts. Concatenation at C-V boundaries should be avoided. In particular, if C is not continuant, the concatenation at C-V boundaries is highly penalized.
- The discontinuities between vocalic sounds often cause degradation of speech quality.
- There will be a join in each and every unit. Certain phonemes do not join well in this way.

The concatenation between two non-consecutive units degrades the resulting speech quality. The number of this kind of concatenations performed should be as low as possible. To improve the naturalness of the synthesized speech, the length of basic units is increased for concatenation, from demi-phones, diphones, triphones, syllables and words to variable length units. The term non-uniform unit means a unit of variable length. An extreme example of maintaining naturalness is the use of pre-recorded speech for all cases. A step beyond this is word or phrase level concatenation of speech segments from pre-recorded utterances. However, we also wish to increase flexibility and therefore turn to concatenating together smaller-sized units. A decrease in unit length must be accompanied by an increase of context.

A non-uniform unit includes long-span units and a phoneme unit to cope with spectral variations having longer period than the basic units. Non-uniform units were carefully designed to take care of the co-articulation problem for different combinations. Since the amount of speech data is restricted, the longer a unit is, the fewer variants there are. The non-uniform unit inventory is always a tradeoff between size and quality.

During synthesizing, non-uniform and longer units are used when they are available. The smallest unit is adopted only when no suitable larger units can be used.

### 3.2. Weakness of Viterbi decoding in speech synthesis

In the so-called uniform unit selection system, Viterbi search is widely used to go through the whole lattice. The process of selecting an optimal sequence of units from a voice database has focused on searching unit graphs with a distance metric consisting of two costs: a target and a concatenation cost. Target costs can incorporate information about phonological environment, spectral measures, and prosody measures. Concatenation costs can incorporate information about spectral continuity and prosody continuity. Calculating target cost $C^t$ and join cost $C^j$, an optimized path is generated from the beginning of lattice to the end of lattice. After the

Viterbi search is complete, the Viterbi path is obtained from the back trace through the graph.

When a very large speech database is used, the number of instances for a certain unit is quite high. It will consume a lot of computing resources to extend each instance. Viterbi search occupies excellent pruning characteristics lent by its dynamic programming formulation. Pruning can be done for each target unit to reduce the complexity.

The Viterbi algorithm makes several assumptions. One assumption is that computing the most likely sequence up to a certain point $t$ must depend only on the observed event at point $t$, and the most likely sequence at point $t - 1$. During Viterbi decoding, a decision is made based on its best predecessor state and the transition from the previous state to the current state. It will not examine later scores. The score of predecessor and transition penalties are calculated only locally. There are chances that the best successor at global level will not be extended. Because of pruning and vulnerability of assumption, the result may not be optimized for the global level.

In order to provide enough variants, the concatenative TTS system always uses a very big database. For such a big speech database, annotation of unit boundary has to depend on automatic tools. Currently the labeling results are still far from perfect. When calculating the join cost of two units, this kind of inaccuracy will not cause trouble for two consecutive units in the same chunk in the voice database. The join cost is zero. Even if there is some offset to the accurate point of the boundary point, the concatenation cost will still be zero. However, for two non-consecutive units in a voice database, e.g., two units from two different words, the join point will not match exactly, even if the unit boundaries are aligned accurately and both units satisfy the context perfectly. The cost will be above zero. This brings in some challenges to unit selection. Any offset to the correct boundaries of both units makes the join cost bigger. This kind of error could lead the suitable unit pairs missed. Such practical issues make the situation worse.

### 3.3. Chunk search at a different layer

The decoding procedure of speech synthesis is a little different from speech recognition. For speech recognition decoding, the boundary of word or longer unit is unclear. To find the word boundary in audio during recognition is impractical, or a similar task to recognition itself. For speech synthesis, fortunately, the whole voice database is preprocessed. It is essentially a closed set. The target sequence has a clear phonetic and prosodic structure after front-end module processing as shown in Figure 2. The vertical lines between units in figure 2 mean boundaries of different layers. Please note that in figure 2 Initial/Final is used as the smallest units in Mandarin just for convenience. In this paper, a phoneme is taken as the smallest unit. The units in different layers are called chunks.

In order to maximize the integrity of the fundamental unit, the new search strategy decodes the chunks layer after layer, from bottom to top.

Given the output from text analysis, all units in different layers can be seen as chunks. From bottom up, they are phonemes, syllables, words and phrases. We take a phoneme as the smallest unit in this paper.

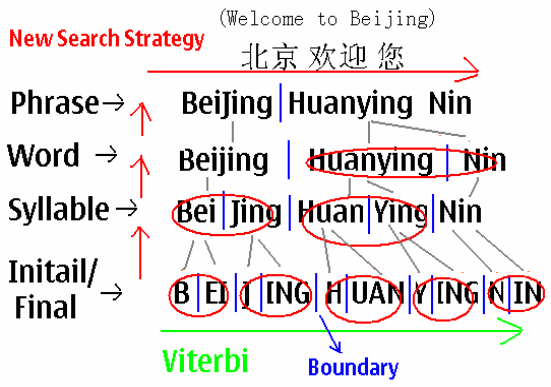


*Figure 2: new search strategy*

During search procedure, in the first round all units at the syllable layer can be decoding separately using Viterbi. These instances in the voice database should have a low join cost. If these syllables exist in the voice database, the value of join cost is zero. If the number of instances is high, pruning will take place, and only those instances which have the most similar or exactly the same context will survive. If the number of instances is less than the N-best requirement, some phoneme sequence will be generated using phonemes from different syllables. This scheme provides solutions for unseen words, or just a new context environment which was not seen in source speech.

After the first round, all top N-bests form the candidate list for the corresponding syllables. The second round for the word layer can be searched out as a syllable round. Candidates for each chunk are from the result of the previous round. Again the output of each chunk forms a candidate for the next layer, phrase.

From the candidates of words, it is easy to search at the phrase level. The results can be seen as the results for the whole sentence.

Using the above procedure, for a target sequence, if an instance of this sequence exists in the source database, this instance will be selected. Basically the maximum length of the target sequence will be selected out, as the non-uniform unit selection has usually done.

During the multilayer search, it is possible to skip some middle layer, e.g., the syllable layer. The whole layer framework depends on the concrete language, or what kind of granularity is wanted.

The new search strategy focuses more on the integrity or quality of the fundamental unit. Inside the chunk, e.g., a word, the accuracy of phoneme boundary annotation is not as sensitive as in the usual search method. The corresponding unit sequence will be selected. Of course the boundary accuracy does effect wav concatenation. It is possible some part of the wav, no matter if it is in time domain, frequency domain or another parameter domain, will be missed. But it will not affect the selection procedure. In case that a long unit sequence is selected, the boundary errors will not be accumulated. Only the boundary error of the beginning and ending unit will be revealed. Furthermore, the units which the beginning and ending unit abut may be silences or pauses. Their relative boundary tolerance is wider than other units'.

The cost of concatenation at points of low power or pauses are relatively low.

During selection procedure, prosody information is not specially considered. It is assumed that prosody is contained in the context implicitly. The prosody structure information, such as prosody phrase boundary, can be considered as a different layer for decoding.

### 3.4. Limitations

This new chunk search at different layers is not time synchronous any more. Several rounds of scanning are needed to make decisions for different layers separately, from bottom to top. For the middle layer, only some candidates of high probability are proposed, the final decisions have to be postponed, and its final selection is subject to the upper layer, until the whole sentence is solved. To start upper layer decoding, state copying is needed to get information of the lower layer. All of the above cause the complexity of search to be increased. Additional memory is also consumed to store the temporary result of the middle layer. Fortunately, the pruning function is still functioning. The load increased is equivalent to adding more phonetic or prosody nodes to the original target sequence. So that it is in linear mode.

Another potential con is that this new search strategy focuses on the integrity of chunks. The join cost between original consecutive units is low. The join cost weight between chunks has to be adjusted to balance so that other non consecutive sequences can still be selected out if the prosody mismatch of the consecutive sequence is significant.

## 4. Building Voice for Blizzard Challenge 2007

### 4.1. Speech database

The speech database is the ATR American English Speech Corpus for Blizzard Challenge 2007[9]. It contains 1032 arctic script utterance, 3617 Basic Travel English Conversation, 1930 news items. In total it has 6579 utterances, around 8 hours. The database came with script and wave files.

### 4.2. Building voice

The whole voice building procedure is just for the general TTS system. The building procedure is done automatically, similar to a multisyn voice build. Using lexicon and text to phoneme rules, the scripts for wav files are turned into phoneme sequences. HTK tools are used to force alignment labeling. The annotation quality is crucial for speech synthesis. For such a large database, manual checking is almost impossible.

Once the segmentation information is ready, the voice creation is straightforward. Pitch mark detection is another important step. Several pitch extraction tools are used to cross check the pitch information to protect from strange values.

The non-uniform unit selection is implemented through search strategy as described in section 3.

### 4.3. Evaluation results

Two sets of voices were submitted to Blizzard 2007. Voice A is from the full dataset while Voice B is from the arctic subset.

The evaluation results from all listeners are given in table 1. The MOS score of Voice A for the full database is about 3.0, while for the ARCTIC data set it is only 2.6. The Word error rate for Semantically Unpredictable Sentences (SUS) test from Voice A is 39%. The Word error rate for SUS test from Voice B is also 39%.

| | MOS | SUS WER |
|---|---|---|
| Voice A | 3.0 | 39% |
| Voice B | 2.6 | 39% |

Table 1. MOS and SUS evaluation results

Voice A has a significantly higher MOS score than Voice B, but surprisingly the SUS scores for the two voices are identical. Using the new search strategy, more focus is put on the integrity of small units. It is supposed that clearer words should be generated. However, the SUS word error rates of two voices are quite high. Further investigation will be done on this issue.

## 5. Conclusions

This paper introduces the entry developed by Nokia for Blizzard Challenge 2007. A new search strategy is proposed to achieve the variable length unit selection with uniform unit. We are trying to retrieve unit sequence from the original voice database as long as possible for the target sequence. Non-uniform unit is not needed to be predefined in voice creation. The voice building procedure is hence less affected.

## 6. References


[1] A. Hunt and A. Black, “Unit selection in a concatenative speech synthesis system using large database”, Proc. IEEE int. Conf. Acoust., Speech, signal processing, pp. 373-376, 1996

[2] Black A. and Taylor P., “Automatically clustering similar units for unit selection in speech synthesis,” in Proc.of Eurospeech, vol. 2, Rhodes, Greece, 1997.

[3] A. J. Clark, K. Richmond, and S. King. 2004. Festival 2 – build your own general purpose unit selection speech synthesizer. In Proc. 5th ISCA Workshop on Speech Synthesis.

[4] Breen A. P., Jackson P., “‘Non-uniform unit selection and the similarity metric within BT’s Laureate TTS system”,Proc. 3rd International Workshop on Speech Synthesis, Jenolan, 1998.

[5] Chu Min; Peng Hu; Yang Hongyun and Chang Eric. Selecting Non-Uniform Units from a Very Large Corpus for Concatenative Speech Synthesizer. ICASSP2001, Salt Lake City, May 7-11, 2001.

[6] King Simon, Portele Thomas, Höfer Florian (1997): "Speech synthesis using non-uniform units in the Verbmobil project", In EUROSPEECH-1997, 569-572.

[7] Yang, J., Zhao, Z., Jiang, Y., Hu, G., and Wo, X. (2006). Multi-tier Non-uniform Unit Selection for Corpus-based Speech Synthesis. Proc. Blizzard Challenge 2006, Pittsburgh, PA, USA

[8] Mark Fraser and Simon King. "The Blizzard Challenge 2007", Proc. Blizzard Workshop (in Proc. SSW6), August 2007, Bonn, Germany.

[9] Jinfu Ni, Toshio Hirai, Hisashi Kawai, Tomoki Toda, Keiichi Tokuda, Minoru Tsuzaki, Shinsuke Sakai, Ranniery Maia and Satoshi Nakamura. "ATRECSS - ATR English speech corpus for speech synthesis", Proc. Blizzard Workshop (in Proc. SSW6), August 2007, Bonn, Germany.