Journal of Computing and Information Technology - CIT 16, 2008, 1, 69–78
doi:10.2498/cit.1001049

An Efficient Unit-selection Method for Concatenative Text-to-speech Synthesis Systems

Jerneja Zganec Gros and Mario Zganec

Research and Development Group, Alpineon, d.o.o., Ljubljana, Slovenia

This paper presents a method for selecting speech units for polyphone concatenative speech synthesis, in which the simplification of procedures for search paths in a graph has accelerated the speed of the unit-selection procedure with minimum effects on the speech quality. The speech units selected are still optimal; only the costs of merging the units on which the selection is based are less accurately determined. Due to its low processing power and memory footprint requirements, the method is suitable for use in embedded speech synthesizers.

Keywords: human language technologies, speech synthesis, corpus-based speech synthesis

1. Introduction

Polyphone or corpus-based concatenative speech synthesis systems usually use extensive speech corpora containing tens of hours of recorded, sampled, segmented, and labeled speech, and use memory of several gigabytes. In such a corpus, each basic speech unit or each speech segment constituting a specific series of basic speech units or polyphones occurs repeatedly in various contexts and with different prosodic characteristics [1].

Limitations in computational processing power and memory footprint used in embedded systems affect the planning of the unit-selection process. Therefore, often HMM-based solutions are applied [2]. Selection of speech units is the part of concatenative or corpus-based speech synthesis that can exert the greatest influence on the speed of the entire speech synthesis process. It is necessary to find a favorable compromise between the size of the speech corpus and the computational complexity of the unit-selection procedure [1]. If the unit-selection procedure is very simplified and thus also very fast, selection of units in a larger speech corpus can be performed in the same amount of time. Oversimplification of the procedure can, however, result in the selection of inappropriate speech units and therefore reduce the speech quality despite using a larger corpus. In contrast, choosing a complex unit-selection procedure can ensure an optimal unit selection, but because of time restrictions this can only be performed on a small speech corpus.

The paper is structured in the following way. In Section 2, unit-selection in polyphone concatenative speech synthesis is introduced as a graph-search problem. An overview of unit-selection methods is presented.

The unit-selection procedure with which we succeeded in accelerating the speed of the procedure without significantly affecting the speech quality is presented in Section 3. This is achieved by simplifying the calculation of the concatenation cost and thus creating the conditions enabling a specific structure of the algorithm to find the optimal path in the graph.

Evaluation of the speed and the speech quality of the proposed unit-selection procedures are presented in Section 4.

2. Unit-selection in Polyphone Concatenative Speech Synthesis

The task of unit-selection procedures is to find the most appropriate speech units in the corpus such that they produce a maximum-quality signal when merged.

Input data that the unit-selection procedure receives from language processing modules in the speech synthesizer are sequences of phonemes to be pronounced, whereby prosodic parameters for the pronunciation of each phoneme are provided. These parameters contain data on the fundamental frequency and duration of the phoneme pronunciation.

Output data that the unit-selection procedure must convey to the module for concatenating speech segments into a speech signal are sequences of specific fragments from the speech corpus called polyphones, or the speech units that the concatenation module will have to merge. These sequences can also be equipped with prosodic parameters for each fragment, which enables the concatenating module to convert the original prosodic parameters from the corpus such that they resemble the desired prosodic parameters to the greatest extent possible.

2.1. Search Graph for Finding the Optimal Sequence of Speech Units

The problem of finding the optimal sequence of recorded units for quality speech signal synthesis can be presented as finding an optimal path in a graph. This kind of presentation clearly demonstrates the problem of selecting speech units and, at the same time, enables the use of recognized procedures for solving this problem. Each vertex of the graph represents a basic speech unit from the speech corpus. The basic speech segments may be allophones, diphones, triphones, or any other basic speech unit. The graph is divided into individual levels. The first level contains the initial vertices; that is, all basic speech units in the speech corpus that correspond to the first basic speech unit in the input character sequence that needs to be synthesized.

The edge between the vertices determines the possibility of merging the basic speech units represented by the connected vertices. In merging speech units, the unit of a higher-level vertex chronologically follows the unit of the lower-level vertex. This is why the edges between the vertices are directed. The vertices are interconnected such that each n-level vertex is connected to all n+1-level vertices.

In this kind of graph, finding the optimal speech unit sequence can be defined as finding the optimal path between any initial vertex in the graph (first level of the graph) and any final vertex in the graph (last level of the graph), whereby the edges between the graph's vertices determine the possible paths.

To start searching for the best path in the graph, criteria expressing the final goal of the speech unit selection must be defined as numeric relations between the data represented in the graph. The final goal of the speech unit selection is the maximum possible intelligibility and naturalness of the synthetic speech. In general, the following criteria have been implemented to make the speech as intelligible and natural as possible:

- The smallest possible number of speech unit concatenations,
- The smallest possible discontinuity of concatenated units at the point of concatenation,

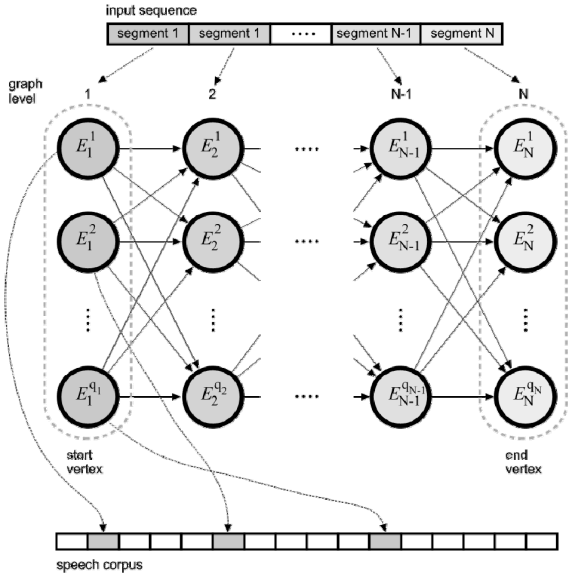

Figure 1. Structure of the graph for finding the optimal speech unit sequence; E_1^i are the graph initial-level vertices, E_N^i are the graph final-level vertices.

- The best fit between the concatenated units' prosodic features and the desired speech prosody.

The first two criteria are evaluated by defining the concatenation cost for every edge between the vertices in the graph, whereas the last criterion is evaluated by defining the cost of fit of prosodic features for every vertex. The cost of an individual path in the graph equals the sum of the costs of vertices through which the path runs plus the sum of costs of all the edges the path contains. The optimal path in the graph is the path with the lowest cost.

2.2. The Cost of Fit of Prosodic Features

The cost of fit of prosodic features expresses the similarity or difference between the prosodic features of a specific speech unit from the speech corpus and the desired prosodic features of the part of the speech signal that the speech unit is to form. The required prosodic features can be determined as in [3] and [4]. The cost of fit of prosodic features usually consists of the weighted result of comparing the speech unit duration and its desired duration, and of the weighted result of comparing the profile of the speech unit basic frequency and the desired fundamental frequency profile. In most cases, the ratio in which the unit's duration and the fundamental frequency profile influence the cost is determined experimentally.

In order to find the optimal speech unit sequence in the graph, the cost of fit of prosodic features is determined for each vertex. It is necessary to calculate this cost for each vertex. Although speech corpora can be very extensive, the calculation of the cost does not constitute a numeric obstacle in finding the optimal path in the graph.

2.3. Concatenation Cost

A speech signal is formed by merging or concatenating speech units from the pre-recorded speech corpus. During the process of merging, audible speech signal discontinuities can occur. We try to evaluate the influence of signal discontinuity on the speech quality through the cost of concatenation.

There are several possible approaches to evaluating the influence of concatenation on the speech quality. The simplest method is to define the cost as "0" for concatenating speech units that directly follow one another in the speech corpus, and to define the cost as "1" for all other speech unit combinations. The use of the cost "0" in units that directly follow one another in the speech corpus is logical because they are already linked together and therefore merging is not necessary. With the use of the cost "1" in units that do not follow one another in the speech corpus, all the concatenations were equally evaluated, regardless of the characteristics of the units being merged. With this kind of concatenation cost, the procedure for finding the optimal speech unit sequence would select the sequence with the smallest number of mergers, regardless of the type of speech units.

A better evaluation of the influence of concatenation on speech quality is achieved if the cost of speech unit merging depends on the allophones that are concatenated. Similar to the previous approach, the cost "0" is defined for the merging of speech units that directly follow one another in the speech corpus.

The most accurate evaluation of the influence of concatenation on speech quality is achieved by taking into account the phonetic features of both units merged when calculating the concatenation cost. In this, the differences in the fundamental frequency, formant frequencies, the amplitude, noise factor, noise spectral features, and so on can be taken into account. However, it should be noted that the use of a large number of parameters requires determination of a large number of weights evaluating the influence of the difference in every parameter on the cost of merging. Determining these weights can be very time-consuming and often includes long-term experiments, empirical solutions, and suppositions. A great deficiency of this method of determining the cost of merging is its numeric complexity. With regard to the fact that concatenation costs are determined individually for every pair of basic speech units from the speech corpus, they are impossible to calculate in advance.

To solve this problem, we propose a compromise solution that is considerably faster, and nonetheless partly takes into account the pho-

netic features of concatenated speech units, is determining the concatenation cost in advance for the individual groups of basic speech units from the speech corpus. In this approach, all the basic speech units in a speech corpus are classified into groups on the grounds of their phonetic features such that the speech units within an individual group phonetically resemble one another to the best extent possible. This is achieved by using clustering techniques. The concatenation costs are calculated and saved in advance for all group combinations.

2.4. Related Work

The optimal path in the graph can be reliably determined by graph traversal whereby all the possible paths in the graph are examined and the best one among them can be selected. The number of possible paths between any initial and final vertex of the graph depends on the number of graph levels and the number of occurrences of the basic speech units in the speech corpus.

Considering that a recording of a speech unit in the speech corpus can occur several thousand times and that input sequences can consist of dozens of basic speech units, it becomes clear that the number of possible paths in the graph is very large. Therefore, not all of the possible paths in the graph are investigated, but various procedures are used to simplify and accelerate the search. Some procedures preserve the optimality of the solution, whereas other sacrifice optimality for the sake of faster operation.

The optimal sequence of speech units is determined by minimizing the cost that reflects a decrease in the quality of the synthesized speech due to spectral differences, differences in the phonetic environment, and mutual merging of speech units. The system that was among the first to use the selection of speech units of variable length was the ATR ν-Talk [5]. In addition to all the parameters used up until then, Hirokawa also suggested the use of prosodic differences in selecting the optimal sequence of speech units [6]. In this approach, synthesized speech is created by concatenating the selected speech units and changing their prosodic features if necessary. The use of information on prosody in the speech unit selection was proposed by Campbell [7], [8].

The procedure for minimizing the sums of both costs employs a search based on dynamic programming or one of its derivatives such as A*. Basic speech units or phonemes are usually used as the basic search units. The existing systems that synthesize speech by concatenating speech segments from an extensive speech corpus use this procedure most frequently. The CHATR speech synthesis system was developed on the basis of these methods [9].

By increasing the number of parameters used in finding or selecting speech segments, the size of the speech corpus has to be large enough. With a sufficiently extensive speech corpus, speech segments that resemble the required input prosodic parameters of the segments can be selected from it. In this case, it is not necessary to change the prosodic features before merging the selected speech segments [10].

Many recent studies that deal with improving the procedures for searching and defining the parameters were taken into account when calculating the cost of segments [11], [12]. Modeling functions for calculating costs is a complex issue.

In the selection of speech segments, search procedures can use additional labeling of segments of various lengths that mark the critical parts where concatenation could result in the potential distortion of the final speech signal [13].

Another approach to speech unit selection is the use of static modeling: FSM [14], DCD [15], GRM [16], [17], and Bulyko [18].

3. The Speech Unit Selection Method with a Simplified Cost of Merging

This section proposes a new and simplified speech unit selection method that is very fast and thus appropriate for implementing the concatenative speech synthesizer in embedded systems.

The basic simplification in this method is that the cost of merging two speech segments depends only on the phonemes that are being joined by merging. If merging is carried out at the center of the phonemes, such as in diphonic synthesis, the cost of merging for each phoneme is defined in the center of the phoneme.

If merging is carried out at the phoneme boundaries, the cost of merging must be defined for all the sequences of two phonemes that can occur in speech. These costs of merging can be defined in advance and are not calculated during the synthesis. In addition to these costs of merging, it is presumed that the cost of merging equals "0" if the segments that are being merged directly follow one another in the speech corpus, regardless of the phonemes joined at the concatenation point.

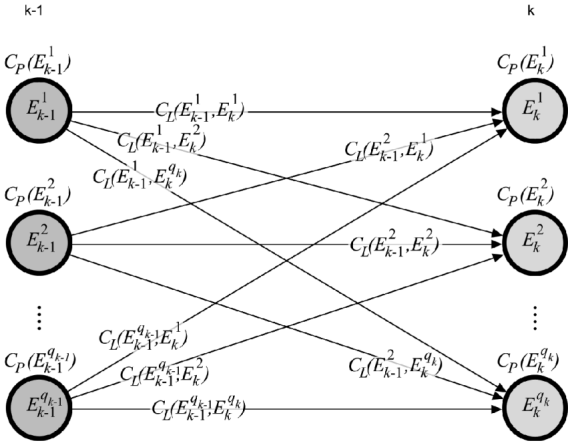

Figure 2. The costs of merging speech segments are defined for the connections between the graph's vertices; k represents the level of the graph. The costs of fit of the prosodic features are defined for all of the graph's vertices.

The graph used in speech unit selection is created as described in the previous paragraph. It comprises N levels, whereby each level corresponds to exactly one basic speech segment in the input sequence that is to be synthesized. At level k of the graph, which corresponds to the speech segment Sk, q_k vertices are located; at level $k+1$, which corresponds to the speech segment S_{k+1}, q_{k+1} vertices are located; and so forth.

Every vertex E_k^i $(1 \leq i \leq q_k)$ at level k of the graph represents a specific recording of the speech segment S_k in the speech corpus. For every vertex E_k^i, the cost of fit of the prosodic features of the corpus speech segment represented by the vertex is also calculated, as well as the required prosodies for the speech segment S_k in the input sequence. This cost is labeled $C_P(E_k^i)$. The vertices are connected by linking every vertex E_k^i $(1 \leq i \leq q_k)$ at level k with all the vertices E_{k-1}^j $(1 \leq j \leq q_{k-1})$ at level $k-1$. The cost of the connection between vertices E_k^i and E_{k-1}^j equals the cost of merging the speech corpus segments represented by the vertices. This is labeled $C_L\ (E_k^j - 1, E_k^i)$.

In finding the optimal path in the graph, it must be established which path between any initial vertex $E_1^i (1 \leq i \leq q_1)$ and any final vertex $E_N^i (1 \leq i \leq q_N)$ of the graph has the lowest cost.

The cost of the entire path is calculated by adding the costs of merging or the costs of edges between the vertices traversed (C_L), and the costs of fit of the prosodic features or the costs of the vertices visited (C_P). Thus, at every level $k\ (1 \leq k \leq N)$ of the graph, only one of the vertices E_k^i $(1 \leq i \leq q_k)$ must be selected, or only one of the speech segments in the speech corpus that will be used in speech synthesis. This vertex is labeled$E_k^{x(k)}$. The cost of the optimal path in the graph can be expressed as:

$$C = \min_{x(1),x(2)\ldots x(N)} \left(C_P(E_1^{x(1)}) + \sum_{k=2}^{N} \Big(C_P(E_k^{x(k)}) + C_L(E_k^{x(k)}, E_{k-1}^{x(k-1)}) \Big) \right).$$

The cost of the optimal path as the function of selecting a vertex $x(k)$ at the individual level of the graph is a decomposable function. If the cost of the optimal path between the graph's initial vertices and the vertex E_k^i at level k of the graph is labeled $C_O(E_k^i)$, and if the cost of the optimal path between the graph's initial vertices and any k-level vertex is labeled C_k, the following applies:

$$C_k = \min_{x(k)} \Big(C_O(E_k^{x(k)}) \Big)$$

and

$$C_O(E_k^i) = C_P(E_k^i) + \min_{x(k-1)} \Big(C_L(E_k^i, E_{k-1}^{x(k-1)}) + C_O(E_{k-1}^{x(k-1)}) \Big).$$

It can be seen that the function of the cost can be defined recursively or that the cost of the path to vertex E_k^i at level k of the graph depends only on the cost of the prosodic fit for vertex E_k^i and on the costs of optimal paths to the vertices of the previous level $(C_O(E_{k-1}^i))$, to which the costs of merging are added.

In optimizing such a function, dynamic programming can be used to find the optimal path in the graph. This method simplifies the search for the optimal path by dividing it into searches for partial optimal paths for every level of the graph.

In practice, the procedure is designed such that four parameters are defined for every vertex of the graph. The first, parameter $I(E_k^i)$, is an index of the basic speech unit in the speech corpus represented by the vertex. This parameter is already defined for the vertex at the start of the procedure, when the graph is created. The second parameter equals the cost of fit of prosodic features $C_P(E_k^i)$, which is also calculated when creating the graph. The third parameter equals the lowest cumulative cost or the lowest cost of the path between any initial vertex and the current $C_O(E_k^i)$ vertex. This cost is calculated during the optimal path calculation procedure. The fourth parameter is an index of the vertex $P(E_k^i)$ from the previous level of the graph located on the optimal path between the initial vertices and the current vertex. This parameter is also calculated during the graph search procedure.

The procedure begins by defining the cost of fit of the prosodic features of the same vertices for the lowest cumulative cost of initial vertices:

$$C_O(E_1^i) = C_P(E_1^i), \quad (1 \le i \le q_1)\,.$$

In the initial vertices, the indicator of the vertex from the previous level of the graph is set to "0" because initial vertices have no precursor. Then the lowest cost of the path to individual vertices at the second level of the graph is defined:

$$C_O(E_2^i) = C_P(E_2^i) + \min_{j=1}^{q_1} \left(C_L(E_2^i, E_1^j) + C_O(E_1^j) \right), \quad (1 \le i \le q_k).$$

In addition, the (j) index of the vertex at the previous level of the graph located on this path with the lowest cost is recorded. This procedure is repeated sequentially for all remaining levels of the graph:

$$C_O(E_k^i) = C_P(E_k^i) + \min_{j=1}^{q_{k-1}} \left(C_L(E_k^i, E_{k-1}^j) + C_O(E_{k-1}^j) \right), \quad (1 \le i \le q; 2 \le k \le N) \qquad (1)$$

The cost of the optimal path is the lowest among the costs of optimal paths to individual final vertices of the graph:

$$C = \min_{j=1}^{q_N} \left(C_o(E_N^j) \right).$$

The optimal final vertex is the final vertex with the lowest cumulative cost.

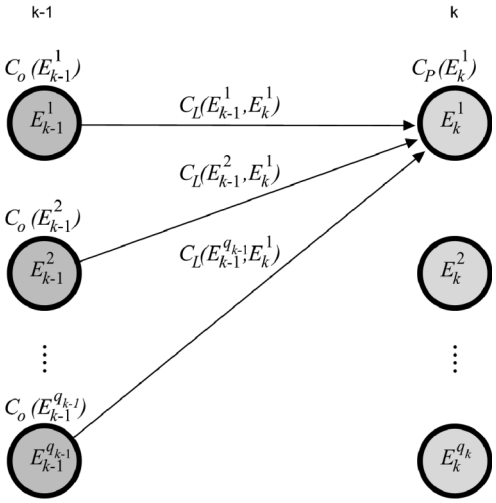

Figure 3. The costs of the optimal path to vertex E_k^i depends on the costs of optimal paths to the vertices at the graph's previous level $C_O(E_{k-1}^i)$, the costs of merging $C_L(E_{k-1}^j, E_k^i)$, and the cost of fit of the prosodic features $C_P(E_k^i)$; k represents the level of the graph.

After the procedure is concluded, the sequence of vertices located on the optimal path is compiled by tracing in reverse the indices of vertices at the previous levels of the graph $P(E_k^i)$ that were saved during the procedure.

With the simplification of the cost of merging introduced in this procedure, the concatenation costs can be determined in advance, so that the cost of merging $C_L(E_{k-1}^j, E_k^i)$ depends only on the type (phonetic group) of speech segments S_k and S_{k-1}. This also means that all the costs

of the edges between the vertices of the graphs E^j_{k-1} andE^i_k are the same for any j and i. This does not apply only if the speech segments represented by vertices E^j_{k-1} and E^i_k, directly follow one another in the speech corpus. In this case, the cost of merging equals 0:

$$C_L(E^i_{k-1}, E^j_k) = \begin{cases} C_L(S_{k-1}, S_k); \\ I(E^j_k) - I(E^i_{k-1}) \neq 1 \\ \\ 0; \\ I(E^j_k) - I(E^i_{k-1}) = 1. \end{cases}$$

$I(E^j_k)$ is the index or the consecutive site of the speech segment represented by vertex E^j_k in the speech corpus.

This means that the calculation of the lowest cost of the path can be further simplified. The recursive equation for calculating the lowest cost of the path to vertex E^i_k is shown in equation (1).

Taking into account the simplifications above, equation (1) can also be expressed as:

$$C_O(E^i_k) = C_P(E^i_k) + \min_{j=1}^{q_{k-1}} \begin{cases} C_L(S_{k-1}, S_k) + C_O(E^j_{k-1}); \\ I(E^j_k) - I(E^i_{k-1}) \neq 1 \\ \\ C_O(E^j_{k-1}); \\ I(E^j_k) - I(E^i_{k-1}) = 1. \end{cases}$$

$C_L(S_{k-1}, S_k)$ is always a positive number. Therefore, equation (1) can also be expressed as:

$$C_O(E^i_k) = C_P(E^i_k) + \begin{cases} \min\Big(C_O(E^J_{k-1}), \\ \min\limits_{j=1}^{q_{k-1}} C_L(S_{k-1}, S_k) + C_O(E^j_{k-1})\Big), \\ if\ \exists J;\ I(E^i_k) - I(E^J_{k-1}) = 1 \\ \\ \min\limits_{j=1}^{q_{k-1}} \Big(C_L(S_{k-1}, S_k) + C_O(E^j_{k-1})\Big), \\ otherwise \end{cases} \quad (2)$$

Because the calculation of the minimum in the equation above does not depend on i, this calculation can be performed only once for all the vertices incident to the same level S_k of the graph:

$$\begin{aligned} C'_O(S_k) &= \min_{j=1}^{q_{k-1}} \Big(C_L(S_{k-1}, S_k) + C_O(E^j_{k-1})\Big) \\ &= C_L(S_{k-1}, S_k) + \min_{j=1}^{q_{k-1}} \Big(C_O(E^j_{k-1})\Big) \end{aligned}$$

Equation (2) can now be expressed as:

$$C_O(E^i_k) = C_P(E^i_k) + \begin{cases} \min\big(C_O(E^J_{k-1}), C'_O(S_k)\big) \\ if\ \exists J;\ I(E^i_k) - I(E^J_{k-1}) = 1 \\ \\ C'_O(S_k) \\ otherwise \end{cases}$$

It can be established that, by using the unit-selection procedure with the simplified concatenation cost described above, only one calculation of the minimum is required for every level of the graph, and only one sum and one comparison for every vertex of the graph. The time required to calculate the optimal path increases almost linearly with the increase in the size of the speech corpus.

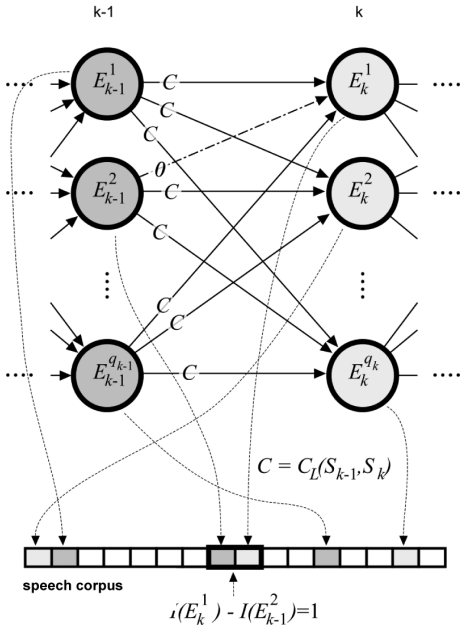

Figure 4. The costs of merging two speech segments directly following one another in the speech corpus equals 0; k represents the level of the graph. The concatenation costs of all other segments depend only on the type (group) of speech segments that are being merged, and are therefore the same for all the connections between two levels of the graph.

4. Evaluation

4.1. Computational Cost

Two versions of embedded concatenative speech synthesis for Slovenian using two different methods for selecting speech units were compared according to synthesized speech quality (subjective evaluation) and computational speed (objective evaluation). The underlying speech corpus, which was used in both experiments, is described in [19]. It consists of 299 sentences, which were selected by a greedy algorithm out of a corpus comprising 2 million sentences. They cover the most frequent collocations, words, quadphones and triphones of the Slovenian language.

The first version used a unit-selection method with a simplified cost of merging described to select speech units described in Section 3, while the second used the widely-used simplified search method for speech unit selection [6], which does not search through all the possible paths in the graph, but limits itself only to the most promising ones. Using the second method, the quality of synthesized speech was slightly lower than the quality of the synthesized speech in the first procedure.

The search time in both speech unit selection methods increases linearly with the length of the utterance that is to be synthesized, and also linearly with the size of the speech corpus, which is an improvement compared to traditional procedures for finding paths in the graph used in speech unit selection of concatenative or corpus speech synthesis. As anticipated, both methods operated increasingly more slowly when increasing the sentence size for which they were seeking the segments required for synthesis.

The simplified search method for speech unit selection is faster than the speech unit selection method with a simplified cost of merging described in Section 3 because it is less complex. As shown in Figure 5, it can find segments for synthesizing shorter utterances twice as fast, and segments for longer sentences four times as fast.

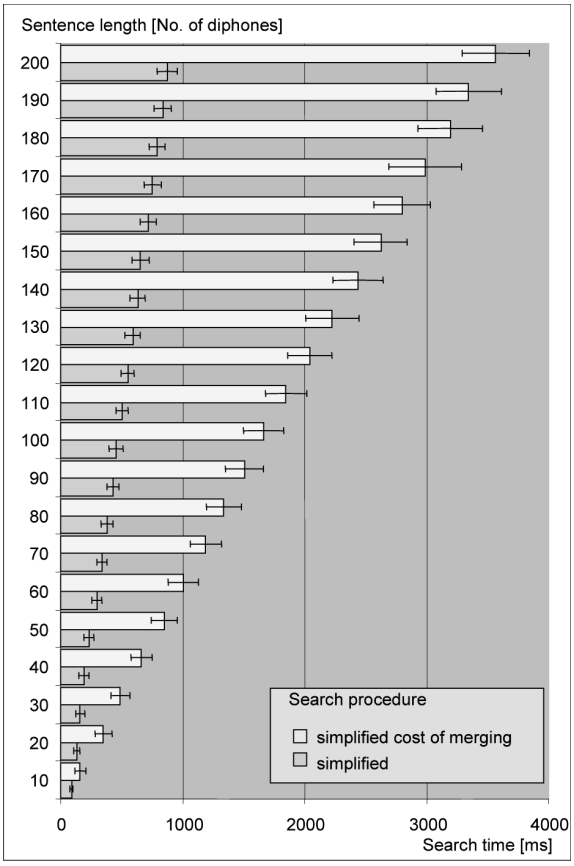

Figure 5. Comparison of computational speed for two unit-selection search methods. As anticipated, the simplified search method for speech unit selection [6] is faster than the speech unit selection method with a simplified cost of merging described in Section 3 because it does not search through all the possible paths in the graph, but limits itself only to the most promising ones. The search speed increases with the length of the sentence for which the procedure must find a suitable speech unit sequence in the speech corpus.

4.2. Subjective Evaluation

Over recent years, various guidelines have been proposed for evaluating the quality of text-to-speech systems. Yet there are still no existing standards for their evaluation, although a number of different methods have been tried and it has been pointed out that the test results they yielded were often inconsistent [20].

The proposed method for polyphone concatenative speech synthesis was tested on an embedded device developed for this purpose [19]. A synthesizer for Slovenian speech was embedded into an automatic system for providing information on honey yields at apicultural observation points.

The adequacy of the resulting synthesized speech was evaluated in terms of acceptability and intelligibility. The experiment was performed in laboratory conditions with 51 test subjects. It was designed according to ITU-T Recommendations P.81 and P.85, describing methods for subjective performance assessment of the quality of voice output devices. The evaluators were selected from a wide range of professional backgrounds, and they were in general not familiar with synthetic voice quality. The test was divided into two sessions, neither lasting more than 20 minutes, in order to reduce the fatigue of the evaluators. Each test speech segment was presented only once.

The first part served to evaluate whether the intelligibility and the quality of the synthetic speech were sufficiently high for a real application of the system in a potential embedded-system application, simulating spoken information on honey yields at apicultural observation points provided by an embedded application in a mobile communicator. The subjects were asked to fill in different application-specific templates based on the information they heard. Each message consisted of a fixed part, which was specific to the task, and a variable part, which was different in all the produced messages. The intelligibility, when spelling errors are ignored, was nearly 100%. Over 95% of the listeners estimated that the embedded TTS system implementation was mature enough for deployment in the given application domain.

In the second part of the test, the performance of the TTS system was evaluated by the same listeners with grades on a five-point MOS (Mean Opinion Score) scale. The listeners were asked to evaluate the overall quality, intelligibility, naturalness, and voice pleasantness.

The overall quality of the speech synthesizer was evaluated as 3.2, or "fair," which corresponds to the overall quality of evaluations of state-of-the-art embedded speech synthesizers for other languages that usually receive grades of approx. 3.5 on the MOS scale.

The majority of the test subjects evaluated the synthetic speech as pleasant and quite natural-sounding, appropriately dynamic and fast and not over-articulated.

5. Conclusions

Limitations in computational processing power and memory footprint used in embedded systems affect the planning of the unit-selection process. This article presents a new method for selecting speech units in polyphone concatenative speech synthesis, in which simplifications of procedures for finding the path in the graph increase the speed of the speech unit-selection procedure with minimum effects on the speech quality. The units selected are still optimal; only the costs of merging the units on which the selection is based are less accurately determined. Further evaluations, in terms of measuring computational speed and assessment of the resulting speech quality by comparing the proposed method to other speech unit selection methods, are planned in future.

Due to its low computational speed and memory footprint requirements, the method is suitable for use in embedded speech synthesizers.

6. Acknowledgment

The work presented in this paper was performed as part of the VoiceTRAN project supported by the Slovenian Ministry of Defense and the Slovenian Research Agency under contract No. M2-0132.

References

[1] M. BEUTNAGEL, A. CONKIE, J. SCHROETER, Y. STYLIANOU, The AT&T Next-Gen TTS System. *Proceedings of the 137th Meeting of the Acoustic Society of America*, (2002).

[2] S. MARTINCIC-IPSIC, I. IPSIC, Croatian HMM-based Speech Synthesis. *Journal of Computing and Information Technology*, **14**(4) (2006), 307–313.

[3] B. VESNICER, F. MIHELIČ, *Evaluation of the Slovenian HMM-based Speech Synthesis System*. Lect. Notes in Computer Science, Springer Verlag, 2006, 513–520.

[4] F. MIHELIČ, B. VESNICER, J. ŽIBERT, E. NOETH, Prosody Evaluation for Embedded Slovene Speech Synthesis Systems. *Inf. MIDEM*, **37**(3) (2007).

[5] Y. SAGISAKA, N. KAIKI, N. IWAHASHI, K. MIMURA, ATR ff-talk speech synthesis system. *Proceedings of the ICSLP'92*, (1992), 483–486, Banff, Canada.

[6] T. HIROKAWA, K. HAKODA, Segment selection and pitch modification for high quality speech synthesis using waveform segments. *Proceedings of the ICSLP'90*, (1990), 337–340, Kobe, Japan.

[7] W. N. CAMPBELL, C. W. WIGHTMAN, Prosodic encoding of syntactic structure for speech synthesis. *Proceedings of the ICSLP*, (1992), 369–372, Banff, Canada.

[8] W. N. CAMPBELL, Prosody and the selection of units for concatenation synthesis. *Proceedings of the 2nd ESCA/IEEE Workshop on Speech Synthesis*, (1994), 61–64, New York, USA.

[9] A. W. BLACK, P. TAYLOR, CHATR: a generic speech synthesis system. *Proceedings of the COLING*, (1994), 983–986, Kyoto, Japan.

[10] W. N. CAMPBELL, Processing a speech corpus for CHATR synthesis. *Proceedings of the ICSP*, (1997), 183-186, Seul, Korea.

[11] T. TODA, H. KAWA, M. TSUZAK, Optimizing Subcost Functions for Segment Selection Based on Perceptual Evaluations in Concatenative Speech Synthesis. *Proceedings of the ICASSP'04*, (2004), 657–660.

[12] J. VEPA, S. KING, Subjective Evaluation of Joint Cost Functions Used in Unit Selection Speech Synthesis. *Proceedings of the INTERSPEECH'04*, (2004), 1181–1784.

[13] S. BREUER, J. ABRESCH, X Phoxsy: Multi-phone Segments for Unit Selection Speech Synthesis. *Proceedings of the Interspeech'04*, (2004), Institute for Communication Research and Phonetics (IKP) University of Bonn.

[14] M. MOHRI, F. C. N. PEREIRA, M. RILEY, The Design Principles of a Weighted Finite-state Transducer Library. *Theoretical Computer Science*, **231**(1), (2000), 17–32.

[15] C. ALLAUZEN, M. MOHRI, M. RILEY, DCD Library – Decoder Library, software collection for decoding and related functions. In AT&T Labs – Research, (2003).

[16] C. ALLAUZEN, M. MOHRI, B. ROARK, A General Weighted Grammar Library. *Proceedings of the Ninth International Conference on Automata (CIAA 2004)*, (2005) Kingston, Canada.

[17] J. R. W. YI, Corpus-based Unit Selection for Natural-sounding Speech Synthesis. PhD. Thesis, Massachusetts Institute of Technology, (2003).

[18] I. BULYKO, M. OSTENDORF, Unit Selection for Speech Synthesis Using Splicing Costs with Weighted Finite State Transducers. *Proceedings of the EUROSPEECH '01*, **2** (2001), 987–990, Aalborg, Danmark.

[19] A. MIHELIČ, J. ŽGANEC GROS, N. PAVEŠIĆ, M. ŽGANEC, Efficient Subset Selection from Phonetically Transcribed Text Corpora for Concatenation-based Embedded Text-to-speech Synthesis. *Inf. MIDEM*, **36**(1) (2006), 19–24.

[20] Y. ALVAREZ, M. HUCKVALE, The Reliability of the ITU-T P.85 Standard for the Evaluation of Text-to-speech Systems. *In Proceedings of the ICSLP'02*, (2002), 329–332, Denver, CO.

Received: May, 2007
Revised: November, 2007
Accepted: November,2007

Contact addresses:

Jerneja Zganec Gros
Alpineon R&D
Language Technologies Group
Ulica Iga Grudna 15
Ljubljana, Slovenia
e-mail: jerneja@alpineon.si

Mario Zganec
Alpineon R&D
Language Technologies Group
Ulica Iga Grudna 15
Ljubljana, Slovenia

JERNEJA ZGANEC GROS is heading the Research and Development Group at Alpineon, Slovenia. Her areas of interest include human language technologies, with the emphasis on speech-to-speech translation, speech recognition and speech synthesis. She is coordinating the largest Slovenian HLT project, VoiceTRAN: http://www.voicetran.org.

MARIO ZGANEC is leading the research group at Alpineon, Slovenia. His areas of interest include human language technologies, esp. embedded speech processing and image processing.