ADMISSIBLE STOPPING IN VITERBI BEAM SEARCH FOR UNIT SELECTION IN CONCATENATIVE SPEECH SYNTHESIS

Shinsuke Sakai[1,2,3], *Tatsuya Kawahara*[3,1], *Satoshi Nakamura*[1,2]

[1] National Institute of Information and Communications Technology, Japan
[2] ATR Spoken Language Communication Research Labs, Japan
[3] School of informatics, Kyoto University, Japan

ABSTRACT

Corpus-based concatenative speech synthesis is very popular these days due to its highly natural speech quality. The amount of computation required in the run time, however, is often quite large and various approaches have been proposed for reducing this runtime computation. In this paper, we propose early stopping schemes for Viterbi beam search in the unit selection, with which we can stop early in the local Viterbi maximization for each unit as well as in the exploration of candidate units for a given target. It takes advantage of the fact that the space of the acoustic parameters of the database units is closed and certain upper bounds of the concatenation scores can be precomputed. The proposed method for early stopping is *admissible* in that it does not change the result of the Viterbi beam search if the upper bounds are properly computed. Experiments show that the proposed methods of admissible stopping effectively reduce the amount of computation required in the Viterbi beam search while keeping its result unchanged.

Index Terms— speech synthesis, unit selection, Viterbi search

1. INTRODUCTION

Corpus-based concatenative approach to speech synthesis has been widely explored in the research community in recent years [1, 2, 3]. In this approach, the best sequence of phone or subphone-sized synthesis units are chosen from a large inventory of units to synthesize speech from the input text through the minimization of the overall cost. The overall cost is often modeled as the weighted sum of target costs and concatenation (or join) costs defined on various features of synthesis units such as spectral shape, intonation contour, and segmental duration. The sequence of units to be concatenated to form the output is usually chosen by some kind of Viterbi algorithm with beam pruning where the quasi-optimal unit sequence is obtained by the repetitions of local score maximization through the dynamic programming principle. The amount of computation is often times very large due to the large size of the unit database that can often contain more than five hours of speech [4]. Various techniques have so far been proposed to reduce the amount of run-time computation, such as caching of concatenation costs [5] and segment preselection based on usage statistics [6].

In this paper, we describe two strategies for reducing the amount of computation in the Viterbi beam search for unit selection, by taking advantage of the prior knowledge about the closed acoustic space of the unit database. We first describe the basic Viterbi beam search algorithm in the next section. In the following two sections, we present two early stopping methods, namely admissible stopping in the local maximization and admissible stopping for the beam, in detail. We report a preliminary experimental result in the succeeding section, followed by the conclusion.

2. UNIT SELECTION WITH VITERBI BEAM SEARCH

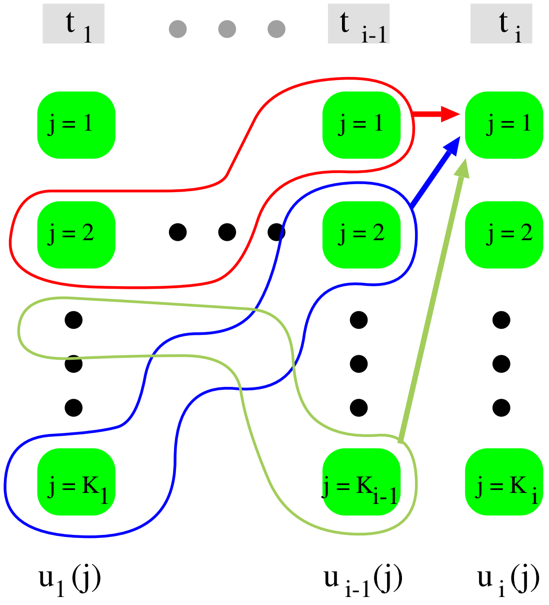

Fig. 1. A schematic diagram that depicts local maximization in the Viterbi algorithm. A gray rectangle labeled t_i stands for the i-th target. Dark circles labeled $j = 1, \cdots, j = K_i$ are candidate units for the i-th target shown above them.

Here we describe the basic Viterbi beam search framework for unit selection that we use for concatenative speech synthesis. Given a sequence of target features $t_1, \cdots, t_I$, we would like to find a sequence of waveform fragments, or units, $u_1, \cdots, u_I$, that maximizes the total score $S(u_1, \cdots, u_I)$. This total score $S(u_1, \cdots, u_I)$ is defined as the sum of target scores over the unit sequence $u_1, \cdots, u_I$ with the sum of concatenation scores over the same sequence added to it,

$$S(u_1, \cdots, u_I) = \sum_{i=1}^{I} L_t(u_i) + \sum_{i=2}^{I} L_c(u_i|u_{i-1}), \qquad (1)$$

1-4244-1484-9/08/$25.00 ©2008 IEEE

Basic Viterbi beam search

(Notation)
$u_i(k)$: k-th database unit for the i-th target.
K_i: the number of database units for the i-th target.
K_θ: the beam width or the number of hypotheses retained at each stage of the iteration.
$S^*(u)$: the score of the hypothesis (the best partial unit sequence) up to the unit u.
$bt(u)$: the predecessor of the unit u determined by the local maximization.

1. Initialization

$S^*(u_1(k)) = L_t(u_1(k))$ for $k = 1, \cdots, K_1$.
Prune the initial set of hypotheses, $\{u_1(k), \cdots, u_1(K_1)\}$, preferring hypotheses with higher scores to keep at most K_θ units.

2. Iteration

Repeat the following for the target indices $i = 2, \cdots, I$:
For all the unit indices $k = 1, \cdots, K_i$ for i-th target:

$$\begin{aligned} S^*(u_i(k)) &= \max_j \{S^*(u_{i-1}(j)) \\ &\quad + L_c(u_i(k)|u_{i-1}(j))\} + L_t(u_i(k)) \\ bt(u_i(k)) &= \arg\max_j \{S^*(u_{i-1}(j)) \\ &\quad + L_c(u_i(k)|u_{i-1}(j))\} \end{aligned}$$

Prune the new set of hypotheses $\{u_i(k), \cdots, u_i(K_i)\}$, and keep at most K_θ hypotheses, preferring hypotheses with higher values of $S^*(u_i(k))$.

3. Termination

$$u_I^* = \arg\max_k S^*(u_I(k))$$

Starting from u_I^*, backtrace $bt(u_I^*)$ recursively, and retrieve the $u_i(k)$'s for $i = 1, \cdots, I-1$ that lead to u_I^*.

Fig. 2. Basic Viterbi beam search for the unit selection.

where $L_t(u_i)$ is the target score for the unit u_i and $L_c(u_i|u_{i-1})$ is the concatenation score for having the unit u_i after u_{i-1}. Maximization of the total score $S(u_1, \cdots, u_I)$ is done efficiently by a Viterbi algorithm. Fig. 1 is a schematic diagram that depicts a local maximization step in the algorithm. The Viterbi algorithm performs global optimization efficiently by repeated local optimizations. However, when the number of candidate units are very large, the amount of computation gets too large to be practical and approximated computation using beam pruning is usually employed. A basic algorithm of this Viterbi beam search is described in Fig. 2.

3. ADMISSIBLE STOPPING IN LOCAL MAXIMIZATION

The number of hypotheses (partial unit sequences up to the preceding target position) to be compared in a local maximization of the Viterbi beam search may be very large and it can often be in the order of thousands. Therefore the situation is very different from typical local Viterbi maximizations such as seen in a left-to-right HMM for speech recognition. We would like to avoid examining all the hypotheses if possible, and a natural expectation is that it may be all right to neglect the hypotheses that are relatively very bad in score.

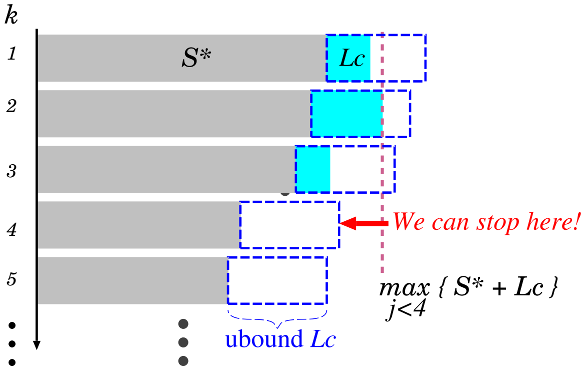

Fig. 3. Cumulative scores $S^*(\tilde{u}_{i-1}(j))$ up to the previous target position sorted in the descending order (dark bars). We see that the sum of the cumulative score and the concatenation score $L_c(u_i(k)|\tilde{u}_{i-1}(j))$ is always smaller than the maximum after a certain value of j.

Now let us assume that we sort the hypotheses that survived the beam pruning in the descending order of the cumulative score $S^*(u_i(k))$. The sorted list of hypotheses is denoted as $[\tilde{u}_i(k)|k = 1, \cdots, \widetilde{K}_i]$, where $\widetilde{K}_i \leq K_\theta$. If the number of hypotheses K_i before pruning was larger or equal to the beam width K_θ, it holds that $\widetilde{K}_i = K_\theta$. If performed in a straightforward way, the number of hypotheses that participate in the local maximization for

$$\begin{aligned} S^*(u_i(k)) = \max_j \{S^*(\tilde{u}_{i-1}(j)) + L_c(u_i(k)|\tilde{u}_{i-1}(j))\} \\ + L_t(u_i(k)) \end{aligned}$$

in the basic Viterbi beam search depicted in Fig. 2 is $\widetilde{K}_{i-1}$. However, as we see in Fig. 3, we can stop in the middle of maximization for some j_0 ($j_0 = 4$ in the figure) with no approximation error, if the cumulative score $S^*(\tilde{u}_{i-1}(j_0))$ is bad enough such that

$$\begin{aligned} S^*(\tilde{u}_{i-1}(j_0)) + \operatorname*{ubound}_{j,k} \; L_c(u_i(k)|\tilde{u}_{i-1}(j)) \\ < \max_{j<j_0} \{S^*(\tilde{u}_{i-1}(j)) + L_c(u_i(k)|\tilde{u}_{i-1}(j))\}, \end{aligned} \qquad (2)$$

where "ubound" in the enequality (2) stands for an upper bound of $L_c(u_i(k)|\tilde{u}_{i-1}(j))$ for all the combinations of the values of j and k, since it holds that $S^*(\tilde{u}_{i-1}(j)) \leq S^*(\tilde{u}_{i-1}(j_0))$ for all $j \geq j_0$ because the list of partial unit sequences up to the $(i-1)$-th stage is sorted in the descending order.

4. ADMISSIBLE STOPPING FOR THE BEAM

In the last section, we discussed an early stopping scheme in the local maximization loop. Now we look at candidate units coming from the unit database at the stage for the i-th target. It would be nice if we can stop in the middle of examining each of $u_i(k)$, when the number of candidate units K_i is much larger than the beam width K_θ.

Let us assume that the set of candidate units retrieved from the unit database for the i-th target, $[u_i(1), \cdots, u_i(K_i)]$, is sorted in the descending order of the target score, $L_t(u_i(k))$. Let us also assume that candidate units associated with the best past partial unit

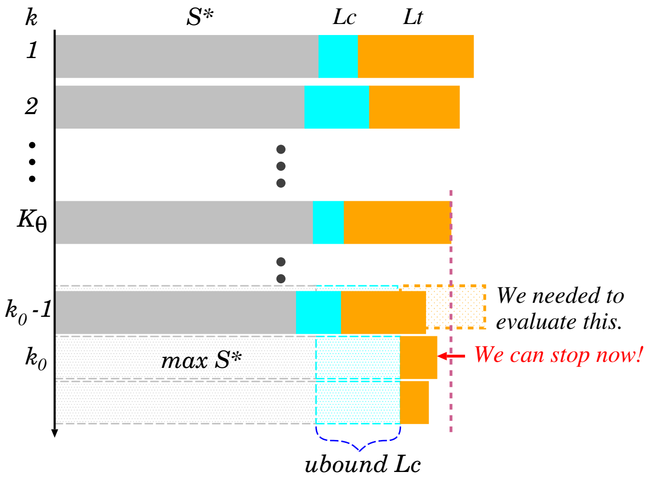

Fig. 4. Units with local maximization done are stored in the descending order of the cumulative score $S^*(\tilde{u}_i(k))$, which is the sum of the previous cumulative score S^*, the concatenation score L_c, and the target score L_t.

sequences are stored in the ordered list $[\tilde{u}_i(1), \cdots, \tilde{u}_i(k)]$ in the descending order of new cumulative scores up to the current target position, $S^*(\tilde{u}_i(1)), \cdots, S^*(\tilde{u}_i(k))$, after local maximizations are done up to k-th candidate unit. As we can see in Fig. 4, after we have explored K_θ units in the i-th stage, we can stop if the target score for some k_0-th unit is bad enough such that

$$\max_j S^*(\tilde{u}_{i-1}(j)) + \operatorname*{ubound}_{j,k} L_c(u_i(k)|\tilde{u}_{i-1}(j)) + L_t(u_i(k_0)) < S^*(\tilde{u}_i(K_\theta)), \qquad (3)$$

since $L_t(u_i(k))$ for $k > k_0$ are all smaller than or equal to $L_t(u_i(k_0))$.

The modified Viterbi beam search algorithm that incorporates the two admissible stopping schemes described in this section and the last section is depicted in Figures 6 and 5.

5. EXPERIMENTS AND RESULTS

We implemented the two admissible stopping methods presented in the last sections in our concatenative speech synthesis system [7, 8]. The unit database was developed using the speaker SLT of the CMU Arctic speech databases [9]. It is spoken by a female speaker of American English and consists of 1132 utterances. The total duration is roughly 50 minutes. The target and concatenation models were all trained using this database. The target score for each unit is a sum of spectral, duration, and F_0 target scores coming from the probabilistic target models described in [10, 8]. Concatenation scores are computed using conditional Gaussian-based models described in [7]. The spectral feature parameters used in the target and concatenation models were both 8-dimensional feature vectors obtained by applying PCA to 14 MFCC coefficients. For the modeling of F_0 and duration targets, fundamental frequencies and durations in seconds were directly used without any transformations.

In the current experiment, we chose to use the maximum values of concatenation scores given by conditional Gaussian-based concatenation models for each of the all possible phone pairs as the upper bounds of concatenation scores. Therefore, in our implementation with 50 phones, they are stored in a table with 50 × 50 entries.

Local maximization with admissible stopping

1. Initialization

Hypotheses (partial unit sequences) up to the $(i-1)$-th stage are listed in the descending order of cumulative score $S^*(\tilde{u}_{i-1}(j))$.
Set $j_{max} = none$, and $score_{max} = -\infty$.

2. Iteration

Starting from $j = 1$, repeat the following for $j = 1, \cdots, \widetilde{K}_{i-1}$ until $S^*(\tilde{u}_{i-1}(j))$ is bad enough such that

$$S^*(\tilde{u}_{i-1}(j)) + \operatorname*{ubound}_{j,k} L_c(u_i(k)|\tilde{u}_{i-1}(j)) < score_{max} :$$

```
if $S^*(u_{i-1}(j)) + L_c(u_i(k)|\tilde{u}_{i-1}(j)) > score_{max}$,
then   $score_{max} = S^*(u_{i-1}(j)) + L_c(u_i(k)|\tilde{u}_{i-1}(j))$,
   and   $j_{max} = j$.
```

3. Termination

$$S^*(u_i(k)) = score_{max} + L_t(u_i(k))$$
$$bt(u_i(k)) = u_{i-1}(j_{max})$$

Fig. 5. Local maximization loop for Viterbi beam search with admissible stopping.

As a preliminary experiment to know the effectiveness of the proposed method, we ran the synthesis system and collected a few statistics on the run-time behavior of the search module of the system.

Table 1. Occurrences of the admissible stoppings in the Viterbi maximizations with various beam widths. The input text was *"Yes, I'd like to leave in the morning."*

beam width	# all loc max	# add stop (%)	# all hyps (x 1000)	# hyps exam (x 1000) (%)
2000	37,142	36,927 (99.4)	45,659	13,051 (28.6)
500	27,817	21,970 (79.0)	13,582	6,377 (47.0)
200	20,334	11,179 (55.0)	4,067	2,599 (63.9)
50	12,061	3,570 (29.6)	603	504 (83.6)

Table 1 shows the number of all local Viterbi maximizations that occurred in synthesizing an utterance (column 2) and the number of Viterbi maximizations that were terminated in the middle by the proposed admissible stopping method (column 3) with its proportion to the number of all Viterbi maximizations (in parentheses). It also shows the number of all the hypotheses to be examined in the Viterbi maximization (column 4) and the number of hypotheses actually examined before the maximization was terminated in the middle (column 5). From the table, we see that the almost all of the Viterbi maximizations were terminated by admissible stopping when the beam is the loosest (2000) and only 28.6% of all the hypotheses actually participated in the local maximizations. When the beam width is very narrow (50), we still see that almost one third of the Viterbi maximizations were admissibly stopped and the number of hypotheses to be examined was still reduced to 83.6%.

Table 2 shows the number of all the units retrieved from the database while synthesizing an utterance (column 2) and the number

Viterbi beam search with admissible stoppings

(Notation)
$u_i(k)$: k-th database unit for the i-th target.
K_i: the number of candidate units from the database for the i-th target.
K_θ: the beam width or the number of hypotheses retained at each stage of the iteration.
$bt(u)$: the predecessor of the unit u determined by the local maximization.

1. Initialization
Retrieve the set of units for the first target from the unit database. Sort them in the descending order of the target score, yielding a sorted list of units $[u_1(1), \cdots, u_1(K_1)]$.

Set $S^*(u_1(k)) = L_t(u_1(k))$ for $k = 1, \cdots, K_1$.

Prune the initial hypothesis list $[u_1(2), \cdots, u_1(K_1)]$, preferring hypotheses with higher scores, to keep at most K_θ units.

2. Iteration
Repeat the following for the target indices $i = 2, \cdots, I$:

Retrieve the set of units for the i-th target from the unit database and sort them in the descending order of target score, yielding a sorted list of units, $[u_i(1), \cdots, u_i(K_i)]$.

Starting from $k = 1$, repeat the local maximization procedure shown in Fig. 5, keeping the new hypotheses in the sorted list $[\tilde{u}_i(1), \cdots, \tilde{u}_i(k)]$, for unit indices $k = 1, \cdots, K_i$. Stop, however, if $k > K_\theta$ holds and the inequality

$$\max_j S^*(\tilde{u}_{i-1}(j)) + \operatorname*{ubound}_{j,k} L_c(u_i(k)|\tilde{u}_{i-1}(j)) + L_t(u_i(k)) < S^*(\tilde{u}_i(K_\theta)).$$

holds.

Prune the list of new hypotheses up to i-th target, $[\tilde{u}_i(1), \tilde{u}_i(2), \cdots]$, to keep at most K_θ units.

3. Termination

$$u_I^* = \arg\max_k S^*(\tilde{u}_I(k))$$

Starting from u_I^*, backtrace $bt(u_I^*)$ recursively, and retrieve the $\tilde{u}_i(k)$'s for $i = 1, \cdots, I$ that lead to u_I^*.

Fig. 6. Viterbi beam search with admissible stopping for the unit selection.

of units actually examined before the early termination by admissible stopping for the beam (column 3) and its proportion in percent. From the table, we see that the number of units to examine was effectively suppressed by the proposed admissible stopping method. We also see that its effect gets greater when the beam width gets narrower, which is expected from Fig. 4.

Table 2. Effects of the admissible stopping for the beam with various beam widths. Input text is the same as Table 1.

beam	# all units	# units examined (%)
2000	37,545	37,142 (98.9)
500	37,545	27,817 (74.1)
200	37,545	20,334 (54.2)
50	37,545	12,061 (32.1)

6. CONCLUSION

In this paper, we proposed two methods of admissible stopping for the Viterbi beam search in the unit selection for concatenative speech synthesis systems that reduce computation in Viterbi beam search without changing the result. One is the admissible stopping in the local maximization, which can terminate the maximization loop in the middle, and the other is the admissible stopping for the beam which makes it possible to disregard the database units with bad target scores without introducing any approximation error. We confirmed the effectiveness of the two admissible stopping methods by experiment. In the current experiment, we used the maximums of concatenation scores for each phone pair as the upper bounds of the concatenation scores. If we know that the subset of units to be retrieved as candidates are more specific than monophones (phones without conditions on surrounding context), such as some kind of tied triphones or biphones, we may be able to prepare tighter upper bounds that give stronger constraints.

7. REFERENCES

[1] A. Hunt and A. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *Proc. ICASSP '96*, 1996, pp. 373–376.

[2] E. Eide et al., "Recent improvements to the IBM trainable speech synthesis system," in *Proc. ICASSP 2003*, 2003, pp. I–708–I–711.

[3] M. Chu, H. Peng, Y. Zhao, Z. Niu, and E. Chang, "Microsoft Mulan – a bilingual TTS system," in *Proc. ICASSP 2003*, 2003, pp. I–264–I–267.

[4] H. Kawai, T. Toda, J. Ni, M. Tsuzaki, and K. Tokuda, "Ximera: A new tts from atr based on corpus-based technologies," in *Proc. ISCA 5th Speech Synthesis Workshop*, 2004, pp. 179–184.

[5] M. Beutnagel, M. Mohri, and M. Riley, "Rapid unit selection from a large speech corpus for concatenative speech synthesis," in *Proc. EUROSPEECH'99*, pp. 607–510.

[6] W. Hamza and R. Donovan, "Data-driven segment preselection in the ibm trainable speech synthesis system," in *Proc. ICSLP 2002*, Denver, 2002, pp. 2609–2612.

[7] S. Sakai and T. Kawahara, "Decision tree-based training of probabilistic concatenation models for corpus-based speech synthesis," in *Proc. Interspeech 2006*, Pittsburgh, PA, Sept. 2006, pp. 1746–1749.

[8] S. Sakai and H. Shu, "A probabilistic approach to unit selection for corpus-based speech synthesis," in *Proc. Interspeech 2005*, Lisbon, Portugal, Sept. 2005, pp. 81–84.

[9] J. Kominek and A. Black, "The CMU ARCTIC speech databases for speech synthesis research," Tech. Rep. Tech. Rep. CMULTI-03-177, Language Technologies Institute, CMU, 2003.

[10] S. Sakai, "Fundamental frequency modeling for speech synthesis based on a statistical learning technique," *IEICE Transactions on Information and Systems*, vol. E88-D, no. 3, pp. 489–495, 2005.