Unit Selection for Speech Synthesis Using Splicing Costs with Weighted Finite State Transducers

Ivan Bulyko and Mari Ostendorf

Department of Electrical Engineering,
University of Washington, Seattle, WA 98195. USA
{bulyko,mo}@ssli.ee.washington.edu

Abstract

In this paper we describe how unit selection for concatenative speech synthesis can be implemented efficiently for sub-phonetic units using weighted finite state transducers (WFST). We also introduce splicing costs as a measure to indicate which unit boundaries are particularly good or poor joint points. Splicing costs extend the flexibility offered by the unit selection paradigm. Through a perceptual experiment we demonstrate an improvement in speech quality achieved by using splicing costs during unit selection.

1. Introduction

Recent improvements in limited-domain synthesis have been in the context of unit-selection concatenative synthesis, with a focus on methods for combining whole phrases and words with subword units for infrequent or new words [1]. Weighted finite state transducers (WFST) have been used for various tasks associated with text-to-speech synthesis, including text analysis, text-to-phoneme conversion, prosody prediction, and unit selection, which motivates a unified WFST architecture, where transducers representing each individual module are cascaded allowing efficient application.

The focus of this paper is on a flexible and efficient implementation of unit selection for concatenative speech synthesis under the WFST architecture. We improve the flexibility of the traditional unit selection procedure by introducing a *splicing cost* in addition to a concatenation cost. The splicing cost is a measure of potential perceptual discontinuity that a given unit may incur when a splice is made at its boundary irrespective of the adjoining unit. We conducted a perceptual experiment showing that the output quality of synthetic speech can be improved by using splicing costs for unit selection.

We also propose an approach to efficiently representing the unit selection database in a form of a WFST. Our implementation supports splicing costs along with target and concatenation costs while constraining the number of links to grow linearly with respect to the number of units in the database.

The rest of the paper is organized as follows. We will begin in section 2 with a review of previous work in the area of unit selection for synthesis. The motivation behind splicing costs will be presented in section 3. Our WFST implementation of the unit selection module will be described in section 4, followed by experiments in section 5. In section 6 we summarize the key advances and outline future work.

2. Background

Recently, a growing amount of attention in speech synthesis research has been drawn toward unit selection methods, based on using dynamic programming to search for speech segments in a database that minimize some cost function [3, 4, 5]. The cost function is designed to quantify distortion introduced when selected units are modified and concatenated. Typically there are two components to the unit selection cost function: the *target cost*, which is an estimate of distance between the database unit and the target, and the *concatenation cost*, which is an estimate of the distortion associated with concatenating units that were not originally spoken in sequence.

Target and concatenation costs have mostly focused on segmental distortion, and have included linguistically motivated distances based on phonetic categories [6] and/or spectral distances [5, 7]. In this work we constrained the search space of candidate units by using the decision tree clustering procedure as described in [8]. Thus only units in the appropriate cluster were considered for a given target. Within each cluster, units were assigned a target cost based on their distance to the cluster mean. Concatenation costs are typically computed as Euclidean or Mahalanobis distance between spectral features representing boundary frames of the corresponding units. Motivated by work described in [9], we used a line spectral frequencies (LSF) representation of frames at unit boundaries for computing the concatenation costs.

Many areas of language and speech processing have adopted the weighted finite-state transducer (WFST) formalism [10, 11], because it supports a complete representation of regular relations and provides efficient mechanisms for performing various operations on them. Relations are represented by the states and arcs specified in the WFST topology, where the arcs carry input and output labels and may have weights (costs) assigned to them. Given these weights, the application of the WFST entails finding the best path (i.e. path with the least cost) through the network. Transducers can be cascaded by means of composition, or their functionalities can be combined by the union operation.

The unit selection database can be efficiently represented in the form of a weighted finite-state transducer (WFST), as it was suggested in [3] and implemented in [12, 2]. The authors in [12] use phones as the fundamental synthesis unit. In order to constrain the number of links in the WFST, they introduce a series of domain-independent intermediate layers of states, where all possible unit transitions are mapped into more general phonetic classes, and transition costs between each pair of classes are computed. In [2] units are domain-dependent words and phrases, and therefore the number of possible concatenation points is relatively small.

Unrestricted TTS demands the use of subword units in the unit selection WFST. Smaller units result in a larger network, which requires more computational power to be constructed. One approach to reduce the computational complexity is to prune the unit database [13]. Alternatively, since computing concatenation costs is the slowest operation, one can precompute and cache concatenation costs between the most frequently used pairs of units [14], or vector quantize the space of units and store a complete distance table between groups of units [5]. In this work we took the vector quantization approach and integrated it into the WFST architecture, as we will explain in section 4.

3. Splicing Costs

Choosing the inventory of units is a subject of ongoing research. Diphone-based systems have been offered for many years. Such systems can produce very intelligible synthetic speech, but tend to sound unnatural due to the limited number of units from which the selection is made. More recently, researchers addressed the challenge of selecting units from a large inventory of phone-sized segments [3, 15]. Systems such as CHATR [15] are capable of synthesizing very natural speech, but fail to do so consistently. One of the reasons for such inconsistent performance is that phone-based systems demand phone boundary concatenations even for the cases where diphone boundaries may produce smoother joints. The AT&T Next-Gen TTS system [5] overcomes this limitation by using half phones for the unit inventory, which allows concatenations to be made at phone boundaries as well as mid-phones. The system allows scaling the concatenation costs at phone boundaries relative to the mid-phone. This makes it possible to tune the system's behavior, gradually changing it between phone and mid-phone concatenation.

Even though Next-Gen offers improved flexibility, it does not provide finer controls (other than the concatenation costs) for deciding in every particular instance whether a phone boundary or mid-phone concatenation is more desirable. For example, there is evidence that stops and fricatives have minimal coarticulation effects [11] and therefore are less likely to have perceived discontinuities at joins at the phone boundaries. Vowels, on the other hand, are found to have smoother concatenations in the middle of the phone. Furthermore, different vowels have different degrees of perceived discontinuity when spliced in the middle of the phone [16]. This evidence motivates an implementation of a more flexible unit selection framework, that would provide separate controls for quantifying the potential perceptual discontinuity at a given boundary, separately from the spectral mismatch between the candidate units.

In this work, we introduce a *splicing cost* to be a measure of the potential discontinuity that a given unit may incur when a splice is made at its boundary. Each unit has two splicing costs : one for each of its left and right boundaries respectively. Since our units are half phones we have splicing costs for both phone boundaries and mid-phones. Controlling splicing costs separately from concatenation costs makes it easier to tune the system parameters and to perform controlled perceptual experiments. Furthermore, the dynamic search at run-time can be made more efficient if the search tree is pruned based on the splicing costs prior to evaluating all possible concatenations.

We hypothesize that splicing cost is inversely related to the spectral change at a given boundary. In other words, unit boundaries where the spectral characteristics in the original recording are changing rapidly are potentially good splicing points. Here we investigated a simple measure of rapid change: the Mahalanobis distance between successive LSF vectors at the splice point. Splicing costs can be efficiently integrated into the WFST architecture, as we will explain in section 4.

4. WFST Implementation

As it was mentioned earlier, we have taken the vector quantization approach to reduce the complexity associated with computing the concatenation costs. Frames at unit boundaries were vector quantized using a 256-vector codebook. The frames were 10 ms wide and contained line spectral frequencies (LSF) of order 14, energy, and F0. Concatenation costs were then computed between each pair of VQ codebook entries by taking the Mahalanobis distance between overlapping vectors at candidate join points. Vector quantization allows concatenation costs to be treated independently of the unit identity and to be represented in the form of a fixed-size fully in-

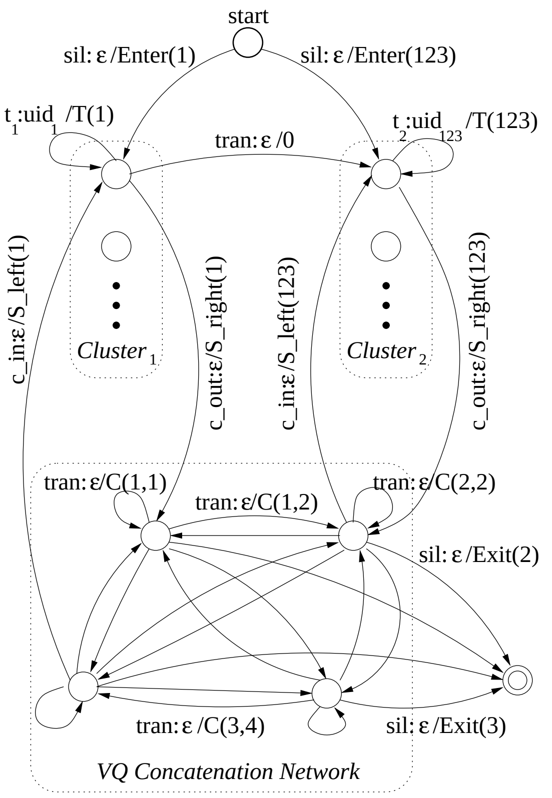

Figure 1: WFST implementation of the unit selection module. Shown are two unit clusters and a VQ concatenation network of size 4. Costs are labeled as follows: $T(u)$ is the target cost for unit u; $C(i,j)$ is the concatenation cost between vector quantized frames i and j; $S\_left(u)$ and $S\_right(u)$ are left and right splicing costs for unit u; $Enter(u)$ is the cost of starting an utterance with unit u; and $Exit(i)$ is the cost of transitioning from VQ frame i to silence.

terconnected WFST, with the number of states equal to the number of codebook entries. Each link transduces a special word "tran" into an empty string ϵ. An example of a small VQ concatenation network is illustrated in the bottom of Figure 1 representing a codebook of size 4. Note that each node has a self loop with a non-zero cost $C(i,i)$ which we define to be equal to the mean distance between all pairs of frames that are mapped into the i-th entry in the codebook. In our implementation two units have a zero concatenation cost only if they occur adjacent in the original recording.

Each unit has a corresponding state in the WFST. In Figure 1 these states are grouped into clusters, representing the leaves of the decision trees that we used to cluster the candidate units. Thus $Cluster_1$ contains all candidate units for the target specification t_1. Each state in these clusters has a self loop that carries out a transduction of a given target t to a unit ID uid uniquely specifying the unit in the database. The cost of this transduction is given by the target cost $T(uid)$, which we define to be the distance

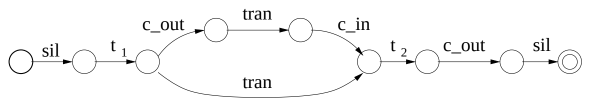

Figure 2: WFST specification of the target

between the unit and its cluster's mean.

Each unit state has two links, incoming and outgoing, that connect it to the VQ concatenation network, where corresponding states represent frames at the unit's left and right boundaries respectively. The costs assigned to these links $S\_left(uid)$ and $S\_right(uid)$ are the left and right splicing costs for the unit. We compute splicing costs as the inverse of the Mahalanobis distance between two consecutive frames at a given boundary.

If two units are adjacent (i.e. have a common boundary) in the original recording, then the concatenation and splicing costs between them are zero, which is implemented by having a direct link from the left unit to its right neighbor with a zero concatenation cost, as shown in Figure 1 between units uid_1 and uid_{123}.

The network has a start state with outgoing links connecting it to each of the unit states. The costs assigned to these links (labeled as $Enter(u)$ in Figure 1) indicate how perceptually suitable a given unit is to be the first in an utterance. For simplicity we made these costs equal to zero for all units. There is one exit state in the WFST which can be reached from any of the states in the VQ concatenation network. Each link leading to the exit state has a cost $Exit(i)$ that measures smoothness of transitioning from a given VQ frame i to silence. We compute this cost by taking the Mahalanobis distance between the frame i and the mean of all frames labeled as silence in the recorded utterances.

Normalization and/or relative scaling of costs in the WFST is a design issue, since the costs are not equally important and the different types of costs can have different average values. In practice, concatenation costs are typically scaled higher than target costs to compensate for higher human sensitivity to smoothness of speech [8]. Through informal listening we found that scaling concatenation and splicing costs such that their means are 10 times larger than the mean target cost gives the best performance.

Given the WFST representation of a unit database described above, the unit selection entails composing this WFST with another WFST representing the target and then finding the best path $BestPath(Target \circ UnitDB)$. The format of target specification is illustrated in Figure 2, where two targets t_1 and t_2 are given. For simplicity, we kept all weights equal to zero, but in principle a target may be assigned a non-zero cost based on the cluster's position in the decision tree. Furthermore, the target specification is flexible to allow alternative prosodic targets and multiple pronunciations.

5. Experiments

The unit database used in our experiments was extracted from the the synthesis component of a travel planning system developed at the University of Colorado [17]. The corpus contained approximately 2 hours of speech ($\approx 150,000$ half phone units) and was automatically segmented. F0 was estimated automatically from the signal by means of Entropic tools. We used Festival TTS system to cluster the units according to the decision tree clustering procedure described in [8].

Fourteen utterances were synthesized using the WFST unit selection approach described in section 4. There were no spectral smoothing or prosodic modification applied to the signal. The target sentences were taken from the same domain, i.e. travel planning, but were produced by a different text generator, so while there was some overlap in the vocabularies, many of the target words were not present in the database recordings. Each utterance was synthesized in two different versions: A) where splice costs are computed as inverse of Mahalanobis distance between two consecutive frames at a given boundary; and B) where all splice costs are zero. In version B we increased concatenation costs by a factor of two in order to maintain the relative importance of target costs at approximately the same level.

We conducted a perceptual experiment, where six subjects, all native speakers of American English, compared the two versions indicating whether one sounds much better/worse (score of 2/-2) or a little better/worse (score of 1/-1) than the other, or if both versions sound about the same (score of 0). The order of the the sentences and of the two different utterances for each was randomized. The subjects were speech researchers but were naive with respect to the system implementation.

The experiment results gave an average score of 1.13 in favor of version A, which included splicing costs. Version A received non-negative scores 95% of the time, while 81% of all version A scores were strictly positive. For two of the utterances all six listeners indicated that version A was much better (score of 2).

6. Discussion

In summary, we have implemented the unit selection module in a form of a WFST, supporting a flexible and efficient framework for overall system integration under the WFST architecture. We have also introduced the use of splicing costs and demonstrated that their use in the unit selection procedure leads to improved quality of output speech. Our approach to computing splicing costs is based on acoustic features and therefore is completely automatic. It is, however, possible that other features (e.g. articulatory, phonetic, or phonological) may contribute significantly to the robustness of splicing costs. In addition, further research is needed on optimizing relative weighting of costs, particularly when prosodic targets are considered as in [2].

7. References

[1] Black, A. and Lenzo, K., “Limited domain synthesis”, In *Proc. of ICSLP*, 2:411–414, 2000.

[2] Bulyko, I., and Ostendorf, M., “Joint prosody prediction and unit selection for concatenative speech synthesis”, In *Proc. of ICASSP*, 2001.

[3] Hunt, A., and Black, A., “Unit selection in a concatenative speech synthesis system using a large speech database”, In *Proc. of ICASSP*, 1:373–376, 1996.

[4] Hon, H., Acero, A., Huang, X., Liu, J., and Plumpe, M., “Automatic generation of synthesis units for trainable text-to-speech systems”, In *Proc. of ICASSP*, 293–296, 1998.

[5] Beutnagel, M., Conkie, A., Schroeter, J., Stylianou, Y., and Syrdal, A., “The AT&T Next-Gen TTS system”, In *Joint Meeting of ASA, EAA, and DAGA*, 18–24, 1998.

[6] Yi, J., and Glass, J., “Natural-sounding speech synthesis using variable-length units”, In *Proc. of ICSLP*, 1167–1170, 1998.

[7] Coorman, G., Fackrell, J., Rutten, P., and Van Coile, B., “Segment selection in the L&H Realspeak laboratory TTS system”, In *Proc. of ICSLP*, 2:395–398, 2000.

[8] Black, A., and Taylor, P., “Automatically clustering similar units for unit selection in speech synthesis”, In *Proc. of Eurospeech*, 2:601–604, 1997.

[9] Wouters, J., and Macon, M., “Control of spectral dynamics in concatenative speech synthesis”, IEEE Transactions on Speech and Audio Processing, 9(1):30–38, 2001.

[10] Roche, E., and Shabes, Y., editors, *Finite-state language processing*, MIT Press, 1997.

[11] Sproat, R., editor, *Multilingual text-to-speech synthesis*, Kluwer, 1998.

[12] Yi, J., Glass, J., and Hetherington, L., “A flexible, scalable finite-state transducer architecture for corpus-based concatenative speech synthesis”, *Proc. of ICSLP*, 3:322–325, 2000.

[13] Donovan, R., “Segment preselection in decision-tree based speech synthesis systems”, In *Proc. of ICASSP*, 2:937–940, 2000.

[14] Beutnagel, M., Mohri, M., and Riley, M., “Rapid unit selection from a large speech corpus for concatenative speech synthesis”, In *Proc. of Eurospeech*, 2:607–610, 1999.

[15] Black, A., “CHATR, version 0.8, a generic speech synthesizer”, System documentation, ATR-Interpreting Telecomunications Laboratories, Japan, March 1996.

[16] Klabbers, E., and Veldhuis, R., “Reducing audible spectral discontinuities”, IEEE Transactions on Speech and Audio Processing, 9(1):39–51, 2001.

[17] Pellom, B., Ward, W., and Pradhan, S., “The CU Communicator: an architecture for dialogue systems”, In *Proc. of ICSLP*, 2:723–726, 2000.