# JOINT PROSODY PREDICTION AND UNIT SELECTION FOR CONCATENATIVE SPEECH SYNTHESIS

*Ivan Bulyko and Mari Ostendorf*

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

**ABSTRACT**

In this paper we describe how prosody prediction can be efficiently integrated with the unit selection process in a concatenative speech synthesizer under a weighted finite-state transducer (WFST) architecture. WFSTs representing prosody prediction and unit selection can be composed during synthesis, thus effectively expanding the space of possible prosodic targets. We implemented a symbolic prosody prediction module and a unit selection database as the synthesis components of a travel planning system. Results of perceptual experiments show that by combining the steps of prosody prediction and unit selection we are able to achieve improved naturalness of synthetic speech compared to the sequential implementation.

## 1. INTRODUCTION

The growing popularity of speech-enabled computer interfaces demands high quality speech output, particularly for telephone applications. The perceived quality of standard general purpose text-to-speech (TTS) systems is not good enough, which forces application developers to use pre-recorded prompts, drastically reducing the text generation flexibility.

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 [16, 3]. Little or no attention has been paid to natural prosody generation, with the assumption that it is accounted for in the phrase-size units. However, as complexity of the domain increases, there is more room for prosodic variability that must be accounted for to achieve natural speech. In this paper, we optimize the synthesizer's performance by unifying the prosody prediction and unit selection under a common framework of weighted finite-state transducers (WFSTs).

In the approach that we propose, the prosody prediction module and the unit database are both represented as WFSTs. During synthesis these WFSTs are composed, resulting in a single transducer that converts a word and phoneme sequence directly into a sequence of database units. As opposed to predicting target prosody first and then searching for units to match that target, our approach effectively makes a "soft" decision about the target prosody and evaluates alternative prosodic realizations of a given utterance, taking advantage of the fact that it is possible to convey essentially the same meaning with prosodically different but yet perceptually acceptable realizations of the same utterance, as evidenced by the variability observed in different readings of the same text [10]. This will introduce some variety into the synthesized speech, which we conjecture will actually improve naturalness.

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. Our approach will be described in Section 3, followed by experiments in Section 4. In Section 5, we summarize the key advances and discuss extensions needed for achieving real-time performance with an unrestricted vocabulary.

## 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 [7, 6, 1]. 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 distortion that the database unit will be subject to when modified to match 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 [16] and/or spectral distances [1, 4]. In this work, for simplicity, the unit concatenation cost function uses a weighted distance based on mel-frequency cepstral coefficients (MFCCs), a variation of which has been found to have a reasonable (0.66) correlation with perceptual distances [15]. In addition, however, we also introduce separate target costs associated with the match between target and database unit prosody in terms of symbolic labels.

Many areas of language and speech processing have adopted the weighted finite-state transducer (WFST) formalism [9, 12], 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 [7] and implemented in [17]. The authors in [17] 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 classes, and transition

costs between each pair of classes are computed. The system also incorporates word-to-phoneme conversion in a WFST module. This module is then composed with the unit selection WFST, allowing conversion of input words directly into a sequence of database units.

Even though the unit database implementation in [17] incorporates both concatenation and substitution costs as part of the WFST, these costs are based entirely on phonetic categories. In most work on unrestricted TTS, acoustic parameters such as F0 and duration are predicted and used in computing the target cost. Alternatively, a recent study [14] has shown that the perceived TTS quality can be improved by including symbolic prosodic labels in the criterion used for unit selection. Use of phonological trees (again, symbolic) in the target specification has also been proposed in [13].

## 3. APPROACH

This section gives details about our approach to prosody prediction and unit selection under a common framework of WFSTs. The process of building prosody prediction and unit selection WFSTs will be described in Sections 3.1 and 3.2 respectively. Then Section 3.3 will explain how these two steps can be tightly coupled by composing the corresponding WFSTs.

### 3.1. Prosody Prediction

As we are building a constrained domain synthesizer, we can expect much of the input to resemble the utterances recorded when collecting data. Therefore, we want our model to precisely describe prosodic structures represented frequently in the training data in terms of so called prosodic "templates". The "templates" are not stored speech waveforms, but rather a symbolic sequence of words and word "slots" that are associated with prosodic labels such as pitch accents and phrase boundary markers. Thus, the templates may or may not correspond to actual phrases in the database, depending on the particular words chosen.

In addition to the "template" prosody model, we want a model that can generalize to previously unobserved input patterns. Hence we need a prosody prediction module, that can be used as a back-off mechanism. The main difference between our approach and the "back-off prosody prediction" suggested by Taylor [13] is that we propose a more dynamic solution by integrating prosody prediction and domain-specific templates using WFSTs.

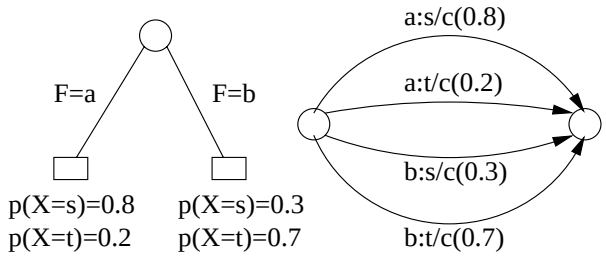


**Fig. 1**. A simple decision tree and its WFST representation, where $F$ is a prediction variable and $\{s, t\}$ are the possible class labels.

Both the template prosody and prosody prediction modules can be represented in the form of weighted finite-state transducers, and both include component modules at the utterance and phrase levels. The template prosody WFST can include more than one prosodic pattern for a particular template, in which case the different patterns are associated with relative-frequency-based weights. (We have found that it is useful to have multiple versions of a template, since our synthesis database does include different prosodic renditions of frequent phrases, presumably for variety since it is not correlated with location in the dialog.)

Prosody prediction is accomplished by building decision (or regression) trees, which can be efficiently compiled into weighted finite-state transducers. A simple decision tree can be represented by a WFST with just two states (a start and an end state) and the number of arcs equal to the number of leaves in the tree times the number of different values that the output variable can take (as illustrated in Figure 1). The costs $c(p(x|leaf))$ in the resulting WFST should reflect the conditional probability distributions that can be computed at each leaf when training the tree. As also suggested in [12], we used $c(p) = -log(p)$ as the cost function in our experiments.

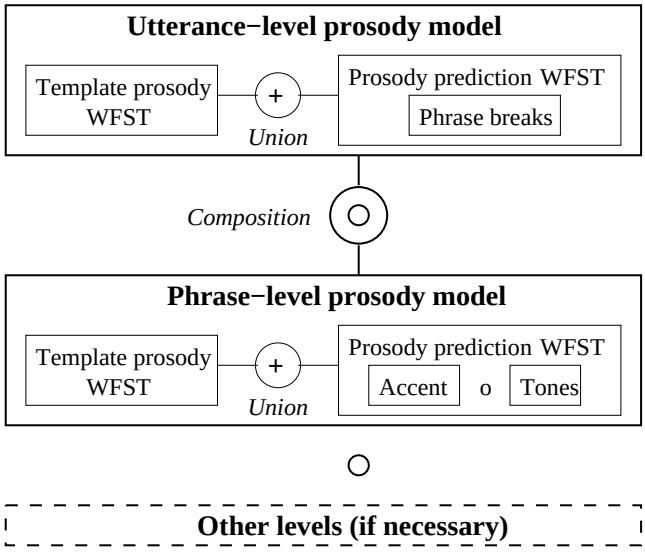


**Fig. 2**. Modular structure of our prosody model, where $\oplus$ indicates union and $\circ$ indicates composition operations on WFSTs.

The overall modular structure of our finite-state prosody model is summarized in Figure 2. Models are generated at two levels: utterance and phrase. At each level two WFSTs are produced: one describes specific prosodic structures in the training data, and the other predicts prosody for unseen cases. The prosody prediction WFST may itself be generated by composing individual transducers, as in the case of accent and tone prediction at the phrase level. The template and prosody prediction WFSTs can be combined into a single transducer by means of the union operation. Finally, the resulting models at each level (i.e. utterance level and phrase level) are composed to form the overall prosody model. The order of terms in the composition corresponds to the order of steps during prosody prediction, i.e. utterance level prosody is generated first and then used as a predictor for the phrase level prosody.

Since our approach assumes that there is allowed variability in a given utterance, no prosodic target will be given zero prosody cost. However, prosodic templates are likely to contain words or phrases that were recorded continuously and therefore incur zero

concatenation costs. High template prosody costs may overrule the zero concatenation cost of longer units when the WFSTs are composed; hence, the prosody costs should be scaled lower. In addition, the costs in the template prosody transducer should also be scaled so that they are (on average) lower than the cost of the decision tree-based prosody prediction, since the templates (when applicable) can presumably model prosody with greater accuracy than the decision tree.

The modular structure allows other levels of prosodic structures (such as word or paragraph levels) to be easily added if desired. In addition, it is straightforward to incorporate transducers associated with gradient phenomena such as pitch range and prominence, though we have not implemented that in this work.

### 3.2. Unit Selection

The units in the synthesis database can be treated as states in the finite state transducer with the state transition cost given by the concatenation cost. In the system implemented for this work, the concatenation cost between units $U_i$ and $U_j$ is the average Mahalanobis distance between overlapping frames: $0.5(d_1 + d_2)$, where $d_1$ is the distance between the last frame of MFCCs in unit $U_i$ and the frame in unit $U_{j-1}$ which precedes unit $U_j$ (as the database was naturally recorded), and, similarly, $d_2$ is computed between the first frame in unit $U_j$ and the first frame in unit $U_{i+1}$ which follows unit $U_i$. This approach is more robust than computing a distance between two consecutive frames, because it does not imply continuity at join points. However, it still can be improved by including F0, energy and amplitude in the distance metric [6], which we plan to implement for our future experiments.

The units in the database can be of arbitrary size. It is, however, important to match the unit inventory to the output of the prosody prediction module in order to satisfy necessary conditions for composing the prosody prediction and the unit selection WFSTs, i.e. the set of output strings from the prosody prediction WFST must be acceptable as input to the unit selection transducer. In the case of limited domain synthesis, many of the responses that a text generator produces are likely to contain words and phrases that were recorded during data collection. These words and phrases can be indexed directly and treated as units in the database, and their sub-word elements can also be used as units. New words can be synthesized by concatenating subword or even subphone units.

### 3.3. WFST Composition

At run time the prosody prediction and the unit selection WFSTs can be composed, resulting in a single WFST capable of transducing target phonological input into a sequence of database units. The composition operation is better than a sequential application of the two transducers because it allows for a wider search space by not making a hard decision when predicting prosody. The relative scaling of costs across these two WFSTs can be tuned according to a given task. Costs for likely prosodic sequences were scaled so that on average they were close to the average concatenation cost. Perceptual experiments may be necessary to determine the optimal scaling.

The modular WFST architecture makes it easy to add new components to the synthesizer. For example, one can design a letter-to-sound WFST that models different pronunciations, which would expand the space of candidate units even further.

**Table 1**. Perceptual experiment results: each entry shows how frequently a given version was rated higher than another (ties not included).

| *Loosing version* | *Winning version* | | |
|---|---|---|---|
| | A | B | C |
| A | *** | 34% | 89% |
| B | 45% | *** | 98% |
| C | 5% | 2% | *** |

## 4. EXPERIMENTS

We implemented the joint prosody prediction and unit selection as the synthesis component of a travel planning system developed at the University of Colorado [8]. The corpus contains approximately 2 hours of speech and was automatically segmented. A trained linguist annotated a subset of the corpus (220 utterances) with ToBI prosodic labels [11]. To alleviate the data sparsity and to lessen the effects of labeling inconsistency we have converted the ToBI labels into a simplified representation, where pitch accents were compressed into three categories: high (H*, L+H*), downstepped (!H*, L+!H*, H+!H*), and low (L*, L*+H). The boundary tones were allowed to maintain all four possible types (L-L%, L-H%, H-L%, H-H%), but only major prosodic boundaries (break index 4) were annotated.

We constructed prosodic templates (as described in Section 3.1) for several types of target sentences common to the text generator, each comprised of a sequence of the compressed ToBI labels. We limited our focus to several types of sentences containing city names in various prosodic contexts. Through informal interactions with the dialog system we found that these types of utterances often had incorrect prosody and could potentially benefit from better prosody prediction. For this feasibility study, we used words as the fundamental units in the database, hence we needed to compute very few concatenation points, and pruning of candidate units was unnecessary.

Fourteen target sentences were synthesized by three different methods: A) no prosody prediction, with unit selection based entirely on the cepstral concatenation costs; B) only one zero-cost prosodic target in the template (the most frequent), with all other alternatives having very high and equal costs; and C) a prosody template that allows alternative paths weighted according to their relative frequency (unobserved events are assigned a fixed and significantly higher cost). The target sentences were chosen so that they do not match any single continuously recorded utterance in the database in its entirety.

We conducted a perceptual experiment, where five subjects, all native speakers of American English, ranked versions A, B and C based on their naturalness. The order of sentences and of the three different utterances for each was randomized. The subjects were speech researchers but were naive with respect to the system implementation and corpus. The rankings submitted by one subject were excluded from the final results because they exhibited a much larger variance than that of other subjects.

The results of our perceptual experiments (as illustrated in Table 1) show that version C was rated the most natural very consistently. Versions A and B did not show consistent preference differences, though A was rated above B somewhat more often, probably because it tended to have smoother (and fewer) concatenations. Two (from the total of fourteen) sentences happened to be the most favorable to version A, probably for this reason. Ex-

cluding these sentences brings the winning rate of C over A up to 98%, suggesting that some attention to signal processing aspects of concatenation would lead to an increased importance of prosodic match.

## 5. DISCUSSION

In summary, we have demonstrated that by combining the steps of prosody prediction and unit selection we can achieve improved naturalness in speech output. The WFST architecture provides a very flexible and efficient framework for implementing joint prediction and selection in a TTS system. The flexibility of WFSTs accommodates the use of variable size units and different forms of prosody generation. The computational efficiency of WFST composition and finding the best path allows real-time synthesis, particularly for constrained domain applications.

While the experiments here are quite limited in scope, we have conducted preliminary tests with the Radio News corpus which indicate that the approach can be applied to unrestricted TTS. 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 [5]. Alternatively, since computing concatenation costs is the slowest operation, one can precompute and cache concatenation costs between the most frequently used pairs of units [2], or vector quantize the space of units and store a complete distance table between groups of units [1]. The latter approach can be smoothly integrated into the WFST architecture by connecting states representing units to an intermediate layer of vector-quantized states with pre-computed transitions between them. Our plans for future work include building a complete TTS system in the travel domain capable of synthesizing out-of-domain sentences.

### Acknowledgments

We wish to thank Bryan Pellom and the Center for Spoken Language Research at the University of Colorado for providing the speech corpus, and Lesley Carmichael for prosodic labeling of the corpus. This material is based upon work supported by the National Science Foundation under Grant No. (IIS-9528990). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

## 6. REFERENCES

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

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

[3] A. Black and K. Lenzo, "Limited domain synthesis," In *Proceedings of ICSLP*, 2:411–414, 2000.

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

[5] R. Donovan, "Segment preselection in decision-tree based speech synthesis systems," In *Proceedings of ICASSP*, 2:937–940, 2000.

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

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

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

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

[10] K. Ross and M. Ostendorf, "Prediction of abstract prosodic labels for speech synthesis," *Computer Speech and Language*, 10:155–185, 1996.

[11] K. Silverman, M. Beckman, J. Pitrelli, M. Ostendorf et. al., "ToBI: A Standard for Labeling English Prosody," In *Proceedings of ICSLP*, 867–870, 1992.

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

[13] P. Taylor, "Concept-to-Speech synthesis by phonological structure matching," *Philosophical Transactions of the Royal Society*, Series A. 356(1769):1403–1416, 2000.

[14] C. Wightman, A. Syrdal, G. Stemmer, A. Conkie, and M. Beutnagel, "Perceptually based automatic prosody labeling and prosodically enriched unit selection improve concatenative speech synthesis," *Proc. of ICSLP*, 2:71–74, 2000.

[15] J. Wouters and M. W. Macon, "A perceptual evaluation of distance measures for concatenative speech synthesis," In *Proceedings of ICSLP*, 6:2747–2750, 1998.

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

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