WFST based Unit Selection for Concatenative Speech Synthesis in European Portuguese

Pedro Carvalho, Isabel Trancoso and Luís Oliveira
L^2F Spoken Languages Systems Laboratory, INESC-ID/IST, Portugal
E-mail: {pedro.carvalho,isabel.trancoso,luis.oliveira}@inesc-id.pt

ABSTRACT

The goal of the current work is to use Weighted Finite-State Transducers (WFSTs) to model the unit selection task, in a concatenative Text-to-Speech system. One of the major difficulties is the design of a perceptually meaningful cost function that weights and combines several features of the available inventory units, matching them to the target information. The WFST approach allows for great flexibility, as well as an elegant formulation for the unit selection problem. Although there is a price to pay in terms of processing power and memory, one of its main advantages is that it allowed us to experiment with different approaches to the problem.

1. INTRODUCTION

Concatenative text-to-speech (TTS) systems generate speech signals by selecting and joining pre-recorded speech units. The length of these units can either be pre-defined (e.g. diphones) or variable. In the latter case, one of the main modules of the TTS system is the unit selection module, whose goal is the selection of the best sequence of units, in order to produce high quality synthetic speech. The work described in this paper deals only with this module, assuming that a previous linguistic processing module has produced the target sequence of phone-like-units (PLU) and corresponding prosodic information (F0, duration and energy). That target sequence will be used as input to the unit selection module. The output (or solution) will be the sequence of inventory units to be concatenated.

The unit selection problem can be addressed in many ways. We followed the approach described in [1], which assigns a given cost to each possible combination of inventory units that can be concatenated to form the desired target sequence. The function that computes this cost may take into account several features of the inventory units, matching them to the target information, in order to evaluate the cost of selecting those units. The most common features used in this function are the phonetic segment context, the target F0 and duration adequacy, and the spectral mismatch of the concatenated units. The design of a perceptually meaningful cost function that weights and combines such features, however, is a very difficult research problem.

The goal of the current work is to use Weighted Finite-State Transducers (WFSTs) to model the unit selection problem, allowing for an elegant and flexible framework for testing several approaches to cost function design. The use of WFSTs for unit selection was previously reported for other languages in [2] and [3]. In our lab, they have also been used in different domains of spoken language processing, including, as far TTS is concerned, grapheme-to-phone conversion [4] and alignment [5].

This paper is divided into six sections: the next one describes our unit inventory; Section 3 discusses the components of the cost function; the following one explains how we used WFSTs for the unit selection task; Section 5 discusses the main implementation issues; we finish by presenting some concluding remarks and directions for future research.

2. UNIT INVENTORY

The unit inventory used in this work was extracted from the BDFALA corpus. This European Portuguese corpus was initially built to serve as the basis for several projects in our lab, from speech recognition to diphone synthesis. It was recorded in a sound proof room, using 16Khz sampling frequency, and includes material read by several non-professional speakers, such as phonetically rich newspaper sentences, proverbs, words carefully selected to include uncommon triphones, dialog extracts, and also logathomes.

To build the unit inventory we have selected material from BDFALA spoken by a single male speaker. The 600 newspaper sentences were selected as a base inventory, to which we later added the 4588 words in order to enlarge our phonetic coverage, and the 1412 logathomes. The unit inventory, totaling 90879 occurrences of 58 different PLUs and amounting to 1h15m of speech, was automatically segmented, labeled and pitch marked. We performed a pitch-synchronous linear prediction analysis, storing 18 LPC coefficients for each pitch mark. Other features like duration, F0, formants F1, F2 and F3, were also automatically computed and stored in a unit inventory database. Although concatenative TTS unit inventories are usually larger and specifically designed for synthesis purposes, for instance allowing a much wider prosodic coverage, this procedure allowed us to quickly obtain an inventory from existing recorded material that should be sufficient for the test environment.

ISBN 1-876346-48-5

3. COST FUNCTIONS

As shown in Eq.1, the cost of selecting and concatenating units *i* and *j* from the inventory, to render target unit *k*, is typically modeled as the sum of two components: the target cost and the concatenation cost.

$$C(u_i,u_j,t_k) = C_T(u_i,u_j,t_k) + C_C(u_i,u_j,t_k) \quad (1)$$

The target cost reflects the adequacy of using the selected units to synthesize the target unit, measuring issues like prosodic and phonetic context adequacy. The concatenation cost reflects the cost of joining those units, measuring issues like spectral and pitch discontinuities. The concatenation cost between adjacent units (recorded consecutively within the same speech file) is zero.

Our first approach to the cost function design problem models the target cost as a weighted sum of three sub-components (Eq.2): the duration mismatch, the pitch mismatch and the distance-to-silence mismatch. Distance-to-silence is our way of including higher level prosodic information, measured as the number of PLUs to reach the next silence. The joint target cost for both units *i* and *j* can be defined as the average of the two target costs:

$$C_T(u_j,t_k) = w_{DT}\cdot D_m(u_j,t_k) + w_{PT}\cdot P_m(u_j,t_k) + w_{DTS}\cdot DTS_m(u_j,t_k) \quad (2)$$

The concatenation cost is also a weighted sum (Eq.3). Similar duration and pitch mismatch components are used, but now the total duration of units *i* and *j* and the mismatch of F0 between both units are used instead. Another component reflects the spectral features mismatch between units, and a final term penalizes the concatenation itself versus following the recorded sequence.

$$C_C(u_i,u_j,t_k) = w_{DC}\cdot D_m(u_{i,j},t_k) + w_{PC}\cdot P_m(u_i,u_j) + w_{Spec}\cdot Spec_m(u_i,u_j) + C_{jump} \quad (3)$$

4. WFST APPROACH TO UNIT SELECTION

Our approach to the unit selection problem consists of the combined use of WFST and additional data structures (indexed by WFST labels). The target sequence and the unit inventory are both modeled by transducers. The first one is represented by a transducer ***T*** such as the one in Figure 1, with sequential numbers as input and PLUs as outputs. For each input label, the prosodic information is stored in the data structure.

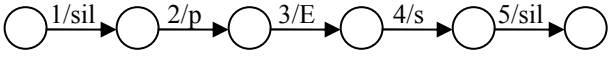

Fig. 1 – Example transducer for the target sequence [pEs].

The inventory units are modeled in a similar way: each recorded file is represented with a transducer that has PLUs as input labels and unit IDs as output labels. Adjacent recorded units have consecutive IDs. The unit inventory transducer ***U*** will contain one transducer for each file, with common initial and final states. If we provide a way for each inventory unit to be allowed to "jump" to another unit with the same PLU, by linking them with an edge with neutral input and output labels (normally represented by ε), then the composition of transducers ***T*** and ***U***:

$$R = T \circ U \quad (4)$$

will yield in ***R*** a transducer containing all possible sequences (or paths) within the inventory transducer that can represent the target sequence. The cost of selecting a specific path can be computed and stored in the weight of each edge of the transducer **R**. Furthermore, transducers ***T*** and ***U*** can also have pre-computed within their edges some of the costs needed to implement the cost function. Finding the best sequence of inventory units to render the target sentence is hence reduced to a simple best-path problem:

$$S = Bestpath(R) = Bestpath(T \circ U) \quad (5)$$

Transducer ***S*** will have in its output labels the inventory unit IDs needed for concatenation, i.e. the solution for the unit selection problem. Here lies one of the advantages of using WFSTs. The problem is formulated in an elegant way, and, as we will see later, there is great flexibility in experimenting with different transducer topologies, different cost functions, etc.

In practice, further refinements are necessary to the topology of ***U***. Figure 2 partially illustrates the topology followed in the present work, in order to decrease the size of ***U*** (number of edges and number of states) and to incorporate more information within its topology. This work used PLUs in context in a similar way as diphones are used in other systems. In this figure, two recorded speech files are modeled by edges starting from the same initial state and following the PLU sequence ([sil,p,E,….] for the first file, [sil,p,a,….] for the second). In the inventory database, units from the first file are numbered 1, 2, 3… and from the second file are numbered 36, 37, 38…

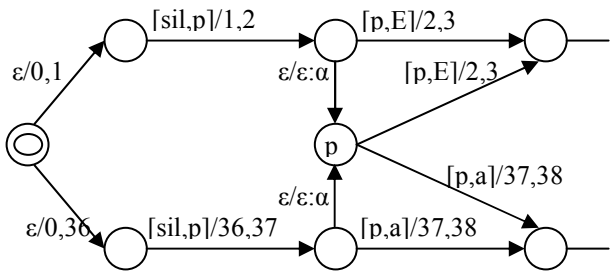

Fig. 2 – Unit inventory transducer ***U*** topology (partial).

Is this topology, each inventory unit can be allowed to "jump" from the original recorded sequence, into another unit of the same PLU (e.g. between units (2) and (37)). The cost of jumping (**α**) is coded into the topology in the edges that link to the "super" state for each PLU ([p] in the figure).

The number of states is proportional to the number of inventory units plus the number of PLUs. The number of edges is proportional to 3 times the number of units.

We opted for using as starting and end states of the inventory transducer all silences in the corpus. Longer sentences are therefore decomposed in several "straight" paths starting and ending with the silence PLU. Although not illustrated, there is an edge from each final silence in each sentence to the initial and final state of this transducer.

We also experimented with other topologies, as the one partially illustrated in Figure 3, that incorporates PLU classes (i.e. vowel, stops, nasal, liquid, fricative, glide, diphthong and silence), allowing the jump state to be decomposed into 3 new states linked by ε/ε edges.

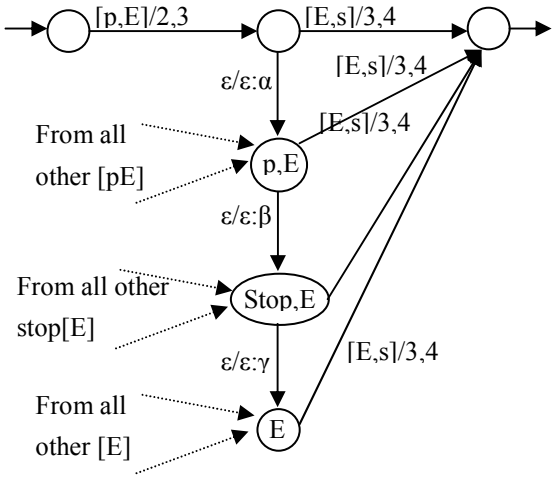

Fig. 3 – Transducer ***U*** topology (detail of jump edges).

The additional "super" states were added to represent each left-context (state marked p,E in the figure) and each left-context class (state marked Stop,E in the figure). Although one of the disadvantages of this topology is that it will increase the number of edges, it also has the advantage of allowing us to model two additional costs (α+β and α+β+γ) for left-context sensitive unit selection. Note that right-context is already assured by the "diphone" nature of the topology and because after a "jump" only edges that match the next diphone are followed. In fact, this is one of the major advantages of using a topology with unit pairs, because it will speed up considerably the composition operation (the number of edges leaving each "super" state is considerable less that in the previous topology).

The pairing of units in ***T*** can be easily achieved by composing the target transducer with another transducer, ***Bt***, that contains the "translation" of PLU to PLU pairs, or computed on-the-fly as a simple unary operation that pairs output labels. Eq.4 must then be re-written as:

$$R = (T \circ Bt) \circ U \tag{6}$$

now that ***U*** contains diphones (PLU pairs) as input labels and unit pairs as output labels.

Since our approach includes on-the-fly computation, it is then necessary to supply the pairs of concatenated units to the cost function for concatenation cost evaluation. This can be achieved by simply pairing the output labels of R, with a similar operation to that performed in the target transducer and therefore Eq.5 can be now expressed as:

$$\begin{aligned} S &= Bestpath[R \circ Bu] \\ S &= Bestpath[((T \circ Bt) \circ U) \circ Bu] \end{aligned} \tag{7}$$

using the transducer that pairs the unit pairs, ***Bu***.

In Eq.1, the cost function can be re-written to include unit pairs and left-context dependent costs:

$$\begin{aligned} C(u_{i-1,}u_i,u_{j-1},u_j,t_k) = C_T(u_i,u_j,t_k) + \\ C_C(u_{i-1},u_i,u_{j-1},u_j,t_k) \end{aligned} \tag{8}$$

In Eq.8, units *i-1* and *j-1* represent the left-context of units *i* and *j*, respectively, that correspond to the same PLU. By observing the concatenation cost sub-component in Eq.8, it is evident the need for pairs of unit pairs and the ***Bu*** transducer. The concatenation cost, depicted in Eq.3, can now be re-written as:

$$\begin{aligned} C_C(u_{i-1,}u_i,u_{j-1},u_j,t_k) = w_{DC} \cdot D_{mm}(u_{i,j},t_k) + \\ w_{PC} \cdot P_{mm}(u_i,u_j) + \\ w_{Spec} \cdot Spec_{mm}(u_i,u_j) + \\ C_{jp}(u_{i-1},u_{j-1}) \end{aligned} \tag{9}$$

$$C_{jp}(u_{i-1},u_{j-1}) = \begin{cases} \alpha & PLU(u_{i-1}) = PLU(u_{j-1}) \\ \alpha+\beta & Class(u_{i-1}), Class(u_{j-1}) \\ \alpha+\beta+\gamma & PLU(u_{i-1}) \neq PLU(u_{j-1}) \end{cases} \tag{10}$$

Here, Eq.10 states that the penalty of making another concatenation in units *i* and *j* depends if their left-context is identical, in the same class or different.

5. IMPLEMENTATION ISSUES

Using our WFST library, **FSTk Tools**, it was possible to develop a modular working environment of four modules:

BuildTarget – receives a transducer representing the target sentence and generates another transducer with PLUs pairs in the output labels, thus implementing the first part of Eq.6. Alternative PLUs may be added at this stage (e.g. stressed/unstressed variants).

FsmCompose – composes the output of BuildTarget with the "unit pairs version" of the transducer inventory ***U,*** thus effectively implementing the second part of Eq.6.

SearchCost – implements the pairing of unit IDs, the cost function computation and the best-path operation, using Best First search or Viterbi search, both with optional pruning. Then it follows the path of the solution and outputs file numbers and time information for the next module.

JustCon – performs the actual concatenation, using time domain waveform zero-cross concatenation with optional energy smoothing.

ISBN 1-876346-48-5

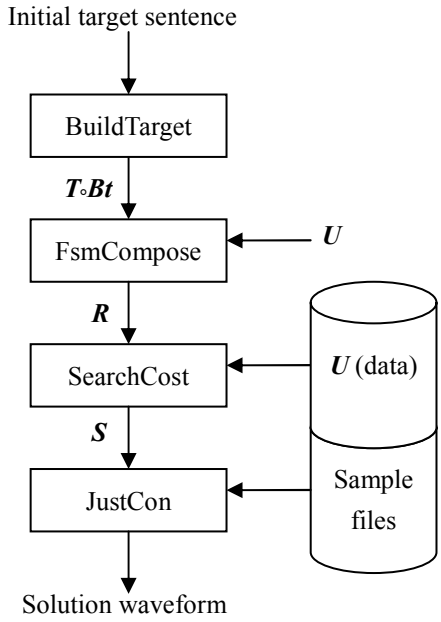

Fig. 4 – Modular development environment

Additional costs were added to the target transducer weights outputted by BuildTarget: the cost of replacing a vowel by its stressed/unstressed counterpart and the cost of substituting glides [j] and [w] by vowels [i] and [u].

As an example of the flexibility of this modular environment, we have developed a 5th module, **FilterCuts**, that can be fitted between BuildTarget and FsmCompose. This module, along with a simple modification in the inventory topology, can modify target transducers to impose a specified number of jumps for some PLU types (i.e. any number of jumps for stops and fricatives, and a predetermined number of jumps for vowels).

One of the major difficulties is to assign perceptually relevant weights to all these factors. As a first approach, the costs were empirically assigned, being one of the issues that we are currently addressing. Other problems arise from the automatic segmentation of the corpus. Errors in automatic grapheme-to-phone conversion and placement of segment boundaries may cause incorrect synthesis or not so natural vowel rendering. Vowel reduction, a very common phenomenon in European Portuguese, is also one of the causes of intelligibility problems. In fact, we found that is better not to allow concatenation in units of small duration or with problems in automatic pitch detection (about 5234 units were classified as non eligible for concatenation).

The memory footprint and processing time are the most important implementation issues. Although it is possible to retain in memory most of the information necessary to use all inventory units, pre-computation of all possible or relevant concatenative costs would exceed current memory limits for personal computers or yield a non-scalable architecture. Thus, it was necessary to make lengthy computations in run-time as the search algorithm goes trough all possible paths within the transducer. On-the-fly computations can help us control the cost in increased memory requirements for all the operations.

To speed up run-time computations, the cost function uses as spectral mismatch a simple Euclidian distance of the pre-computed formants F1, F2 and F3. After finding the solution transducer ***S***, the SearchCost module can use another spectral mismatch function to find out exactly where to cut units *i* and *j*. This cut-point operation at this moment cannot be included in the cost function because it would weight too much on the processing time.

6. CONCLUDING REMARKS

The WFST approach allows for great flexibility, as well as an elegant formulation for the unit selection problem. Although there is a price to pay in terms of processing power and memory, one of its main advantages is that it allowed us to experiment with different approaches.

Planned future work includes: building a larger unit inventory and joining the work described in this paper with our previous efforts in terms of grapheme-to-phone in order to obtain a full TTS system using WFSTs.

7. ACKNOWLEDGMENTS

We would like to thank Diamantino Caseiro for the usage of the WFST toolkit **FSTk**. INESC-ID had support from the POSI program of the "Quadro Comunitário de Apoio III"

REFERENCES

[1] Yoshinori Sagisaka, "Speech Synthesis by rule using an optimal selection of non-uniform synthesis units", *in Proc. of ICASSP'88*, S14.8 pages 679-682, 1988.

[2] M. Beutnagel, M. Mohri and M Riley, "Rapid Unit Selection from a Large Speech Corpus for Concatenative Speech Synthesis," *in Proc. EUROSPEECH 99*, Budapest, 1999.

[3] J. Yi, J. Glass, I. Hetherington, "A Flexible Scalable Finite-State Transducer Architecture for Corpus-Based Concatenative Speech Synthesis", *in Proc. ICSLP 2000*, Beijing, 2000.

[4] D. Caseiro, I. Trancoso, L. Oliveira and C. Viana, "Grapheme-to-phone using finite-state transducers", *in Proc. 2002 IEEE Workshop On Speech Synthesis*, Santa Monica, California, September 2002, 2002.

[5] S. Paulo, L. Oliveira, "Multilevel Annotation of Speech Signals Using Weighted Finite State Transducers", *in Proc. 2002 IEEE Workshop on Speech Synthesis*, Santa Monica, California, September 2002, 2002.