Combining Non-uniform Unit Selection with Diphone Based Synthesis

Michael Pucher(1), Friedrich Neubarth(1), Erhard Rank(1), Georg Niklfeld(1), Qi Guan(2)

(1) ftw. Telecommunications Research Center Vienna, Austria
(2) Siemens Österreich AG
pucher@ftw.at, friedrich@oefai.at, erank@nt.tuwien.ac.at,
niklfeld@ftw.at, qi.guan@siemens.com

Abstract

This paper describes the unit selection algorithm of a speech synthesis system, which selects the k-best paths over units from a relational unit database. The algorithm uses words and diphones as basic unit types. It is part of a customisable text-to-speech system designed for generating new prompts using a recorded speech corpus, with the option that the user can interactively optimise the results from the unit selection algorithm. This algorithm combines advantages of non-uniform unit selection algorithms and diphone inventory based speech synthesis.

1. Introduction

The unit selection method we present in this paper is used for an application that allows a user to create new utterances of high quality using a corpus of pre-recorded utterances of a specific domain, for cases when the speaker is not available. Using this system one can select from a number of candidate unit lists from the corpus and optimise prosodic parameters using the prosody of utterances from the corpus or of a recording by the user (copy-synthesis), optimise concatenation points and add the new utterances to the corpus. The paper will focus on the unit selection module of the system, which selects the k-best unit paths from a relational unit database. Through the combination of words and diphones as basic unit types and a well designed concatenation cost function [1] one can generate high quality synthesized prompts from the corpus.

Our search algorithm is organized in a top-down fashion similar to the phonological structure matching algorithm used in [2]. First, units of word size are considered, if the search fails, the algorithm falls back to units on the diphone or phoneme level. This makes the search considerable fast due to the fact that the concatenation cost is minimal when the selected units originate from the same source, it is guaranteed that the size of actual units used for concatenation are as large as possible. Units of intermediary size, like phrase chunks or syllables, are not directly selected, but entered into the algorithm indirectly. This has the advantage of keeping the algorithm uniform and avoiding phonetically undesired concatenation points, while still minimizing the number of concatenation points by the strong preference for strings of units from the same original signal.

Figure 1 depicts the architecture of the customizable text-to-speech system (cTTS). The unit selection is based solely on automatic transcription of the text and on the annotated speech corpus which is stored in an SQL database [3]. The K-best lists of units are delivered to the concatenation module. Here, and also in the prosody manipulation and signal processing modules the user can manually optimize the search results.

For the annotation of the corpus we use the German version of the Festival speech synthesis system [4][5] and an extended version of the BOMP lexicon [6].
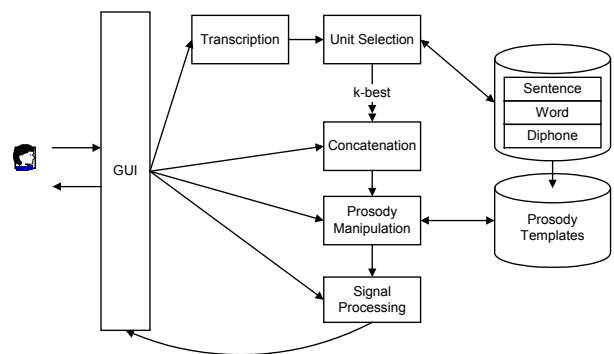

Figure 1: System Architecture of the cTTS system

2. Requirements

In order to apply the proposed unit selection method to a recorded speech corpus, the following requirements must be met:

- The corpus should contain every phoneme in as many phonetic contexts as possible and also at least one instance of diphones (phoneme combinations) with a phonetically complex transition. It should also contain a variety of prosodic patterns.

- The corpus must be fully segmented and annotated.

- The unit selection should use units of maximal size and pattern, and should generate the k-best lists of units.

We assume that the corpus is primarily designed for a specific domain, thus containing a set of the most common prompts, and also the content words which may occur in the planned dialogues. However, it must also contain a list of phonetically relevant combinations of phonemes. This list is appended to the corpus and designed solely on the basis of phonetic considerations.

3. Architecture

Figure 2 shows the design of the database, which is based on the utterance structure of the Festival system and extended by

a prosody template and the concatenation cost table. The concatenation costs are loaded into memory, when the system is started to achieve fast unit selection.
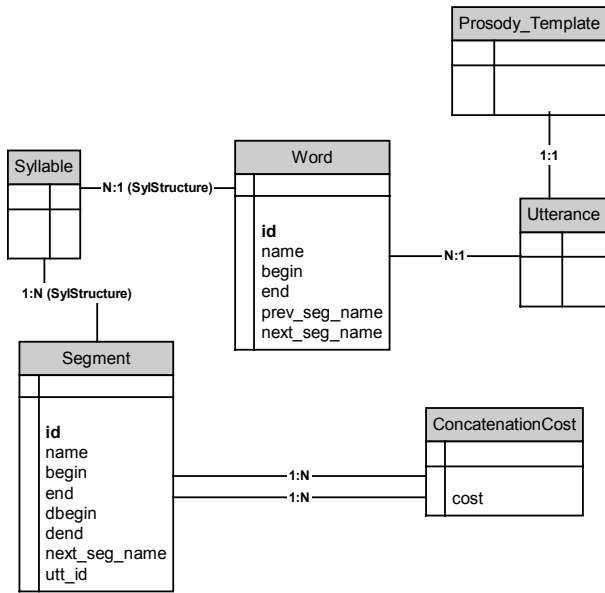

Figure 2: Relational structure of the utterance

4. Cost Function

In order to obtain the concatenation cost between diphones, we calculate the acoustic distance measure from the acoustic features of the diphones. At the moment we use the fundamental frequency, melcepstral coefficients and energy. Each diphone can be joined only with compatible diphones which minimizes the size of the ConcatenationCost table. The concatenation cost for words and phonemes is calculated on the fly.

This is possible, in the case of words, because there are not so many concatenation points. In the case of phonemes it is necessary, because not all possible combinations of phonemes can be stored in the database or loaded into memory. The number of phoneme concatenation points is minimised by the corpus, which contains the most frequently used and also phonetically complex diphones.

A target cost is defined between the database units and the units one wants to synthesize and is calculated also on the fly. For a definition of concatenation and target cost see [1].

5. Diphone Rules

Because the search is performed on the level of diphones, but the symbolic representation refers to phonemes, the annotation for the diphones has to be derived. A unit candidate is defined as a phoneme with its right neighbor. The actual begin and end of the resulting phoneme segment entering the algorithm is re-calculated by a set of rules, thus deriving a dynamic diphone inventory during the setup of the database.

A standard diphone/inventory based system is static and is supposed to be complete. Our system is dynamic and strategic. Dynamic means that the selection of units can be optimized during computation. However, if a diphone candidate is missing from the corpus, the phoneme segments have to be realigned to the original phoneme boundaries, the concatenation costs still guarantee an optimal selection. The burden of completeness is removed from the corpus, but the corpus should contain the phonetically most critical combinations (for example stop-vowel sequences). This is only an indicative demand, if a combination is not found, the algorithm does not fail, but follows a different strategy (see section 6.3).

6. Algorithm

The selection algorithm proceeds in the following order.

6.1. Word Level Search

The main difference between the proposed algorithm and the selection algorithm in [3] is that our search algorithm directly jumps to the diphone level if a word is not found in the database.

First we select all words that are necessary for the utterance. The *utt_id* of the utterance we want to synthesize is set to *0*.

SELECT name, prev_seg_name, next_seg_name, FROM word
WHERE utt_id=0 ORDER BY id

Then we try to find the first, second,… n-th word in optimal contexts by trying the following queries consecutively:

SELECT name, begin, end, FROM word
WHERE utt_id <> 0 AND name=<NAME> AND
prev_seg_name=<PREV_SEG_NAME> AND
next_seg_name=<NEXT_SEG_NAME> ORDER BY id

SELECT name, begin, end, FROM word
WHERE utt_id <> 0 AND name = <NAME> AND
(prev_seg_name=<PREV_SEG_NAME> OR
next_seg_name=<NEXT_SEG_NAME>) ORDER BY id

SELECT name, begin, end, FROM word
WHERE utt_id <> 0 AND name = <NAME> ORDER BY
id

If no units are found for a certain word the algorithm proceeds to the diphone level.

We are aware of the fact, that phonetically speaking lexical words rather than prosodic words are not the optimal candidates for units. However, they are only used as symbolic units for the search algorithm. The cost function guarantees that prosodic words are selected as strings of lexical words, if they are available from the corpus.

6.2. Diphone Level Search

The first query is analogous to the word level. All phoneme-phoneme combinations serving as unit candidates are selected where *dbegin* and *dend* are the borders of the diphone and *begin* and *end* are the borders of the phonemes.

SELECT name, begin, end, dbegin, dend FROM segment
  WHERE utt_id <> 0 AND name=<NAME> AND
  next_seg_name=<NEXT_SEG_NAME> ORDER BY id

6.3. Phone Level Search

If a certain phoneme combination is not present in the corpus, a diphone would be missing for concatenation. Remember that it has to be ensured that no phonetically critical combinations are missing during the corpus setup. The easiest solution is to simply skip such a combination and to use the signal from the neighboring diphones (which do consist of 2 complete phonemes by definition).

Figure 3 shows how this works for a German word "*möglich*" ('possible'). In the upper row, the diphone units are represented. In this example we assume that all combinations are available from the corpus except for the combination of [2:] and [g] and of [C] and silence. The bottom row indicates which segments are selected by the algorithm. An asterisk after the phonetic symbol indicates that only a part of the phoneme is used, combining with the next segment containing the same phoneme. For example 'g l*' means that the full segment of the phoneme [g] is used, but only the begin of the segment of the phoneme [l], which is then followed by a segment 'l* I*', and in turn is truncated at the beginning.

The decision for using the full phoneme segment in case a diphone is missing from the database is not really a fall-back strategy, since the search algorithm does not change, it is rather a directive how to handle a failed search. In effect this allows us to simultaneously use phoneme-sized and diphone-sized units with a built-in preference for the latter. An approach similar to ours uses half-phones as the basic units [7]. However, there it is not guaranteed that critical transitions are selected as diphones, a feature which is explicitly focused in our system.
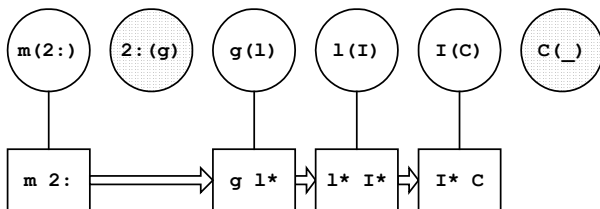

Figure 3: Diphone Unit Selection and corresponding speech segments

6.4. Building the Cost Graph

The search algorithm returns a list of words and diphones, from which a concatenation cost graph is created. At a certain position there are different realizations of a certain unit of a certain type. In a way this strongly resembles phonological structure matching, e.g. the search stops when matching units are found [2].

For each unit we insert a concatenation cost edge to all possible consecutive units.

6.5. Finding k-best paths

The final result of the unit selection procedure need not only give the optimal result but a list of results (the k-best paths through the graph), such that the user can choose between different unit lists, manipulate the units etc. For this purpose we use Eppstein's algorithm, which solves that k-best paths problem for general graphs [8]. The version we use was implemented by Victor Jimenez and Andres Marzal [9].

7. Customisation

The output of the unit selection algorithm is represented as a set of lists of units. Now the user can modify prosodic parameters, listen to the synthesized utterances and choose between different lists via the **G**raphical **U**ser **I**nterface, which is displayed in *Figure 4*.

The k-best lists which are provided by the unit selection can be manually corrected. Each unit segment can be displayed in its original context, the user can manually realign its borders, which is especially useful when larger units are selected or if errors occurred during automatic segmentation (like for the left border of the unit displayed in *Figure 4*). The lists can also be used to tune and evaluate the concatenation and target cost functions. For example one can generate the 10-best lists for an utterance and verify perceptually if they are ordered in decreasing quality.
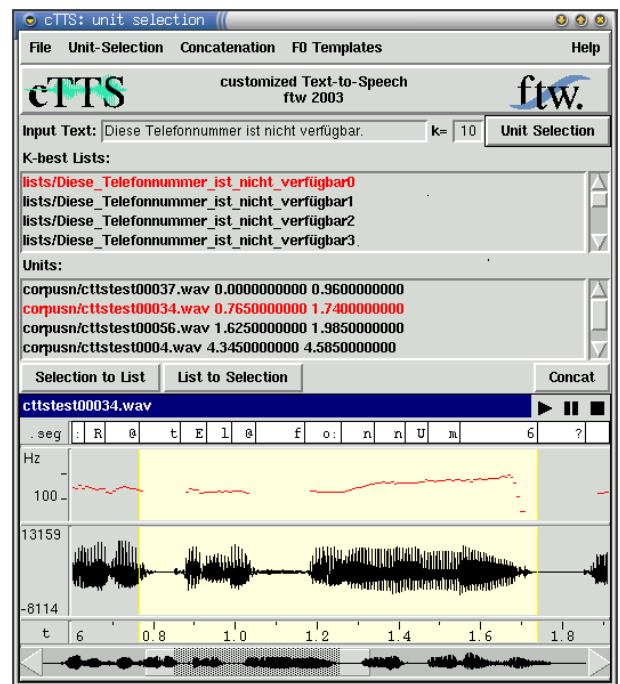

Figure 4: Screenshot of the GUI for manual optimization of the unit selection

Customization also includes parameters for signal processing during unit concatenation. Besides modifying f0 and phoneme duration manually these parameters can also be overlayed from a user recorded utterance (copy-synthesis), or deduced from the prosody template database. Furthermore, the linear prediction based synthesis algorithm allows for

spectral smoothing [10], which can be used for concatenation of units when a spectral mismatch is encountered (e.g., between diphone units [11]).

8. Conclusions

In this paper we have presented a unit selection algorithm which combines the advantages of non-uniform unit selection methods and of diphone inventory based speech synthesis. We have shown that the shortcomings inherent in these methods can be overcome by using a carefully designed corpus, cost functions and a top-down search algorithm.

Further optimisation of the output can be achieved by the user herself, who can manipulate the selection of units, the units themselves and the prosody via a graphical user interface.

In the ongoing evaluation of our system we will measure the search speed and the speech output quality using databases of different size and different weights for the concatenation and target cost functions. We will also develop corpus design guidelines for our system.

9. Acknowledgements

10. References

[1] Andrew J. Hunt, Alan W. Black, "Unit selection in a concatenative speech synthesis system using a large speech database", *Proceedings of ICASSP 96, pp 373-376, 1996*

[2] Paul Taylor, Alan Black, "Speech Synthesis by Phonological Structure Matching**,** *Eurospeech 99, volume 4, pages 1531-1534, Budapest, Hungary, 1999*

[3] Esther Klabbers, Karlheinz Stöber, "Creation of Speech Corpora for the Multilingual Bonn Open Synthesis System", *4th ISCA Tutorial and Research Workshop on Speech Synthesis, Pitlochry, Scotland, 2001*

[4] Alan W. Black. Rob Clark, "The Festival Speech Synthesis System" *http://www.cstr.ed.ac.uk/projects/festival/*, Jan 2003

[5] Gregor Möhler, et.al., "IMS German Festival", *http://www.ims.uni-stuttgart.de/phonetik/synthesis*, July 2001

[6] BOMP, Bonn Machine-Readable Pronunciation Dictionary, *http://www.ikp.unibonn.de/dt/forsch/phonetik/*, 2001

[7] Alistair Conkie, Mark C. Beutngel, et.al, "Preselection of candidate units in a unit selection-based text-to-speech synthesis system", *ICSLP 2000, Bejing, China, October* 2000

[8] David Eppstein, "Finding the k-shortest paths", *SIAM J. Comput., vol. 28, no. 2, pp. 652-673, 1999*

[9] Victor Jimenez, Andres Marzal, "Finding the k-shortest paths", *http://terra.act.uji.es/REA/src/EPPSTEIN-1.1.tgz*, July 1999

[10] Erhard Rank, "Concatenative speech synthesis using SRELP in: Keller et al. (eds.): Improvements in Speech Synthesis", *John Wiley & Sons, pp. 75-86, 2002*

[11] Ester Klabbers and Raymond Veldhuis. "Reducing audible spectral discontinuities", *IEEE Transactions on Speech and Audio Processing, 9(1):39-51, 2001*