Unit selection speech synthesis - an overview

Marc Schröder, DFKI
schroed@dfki.de
Overview

- Concatenative speech synthesis revisited
- Basic idea of unit selection synthesis
- Steps in run-time unit selection
  - target prediction
  - pre-selection of candidates
  - target costs and join costs
  - dynamic programming
  - concatenation of audio segments
- Creating a unit selection voice
  - coverage
  - recording
  - labelling
Concatenative synthesis: Isolated phones don't work

target:  \( \text{w I n t r= d eI} \)

acoustic unit database
(units = \text{phone segments} recorded in isolation)
Concatenative synthesis: Diphones

target:  \textit{w I n t r= d eI} \\
\underline{-w w-I I-n n-t t-r= r=-d d-eI eI-} \\

\textbf{Diphones} = sound segments from the middle of one phone to the middle of the next phone

acoustic unit database units = \textit{diphone segments} recorded in carrier words (flat intonation)
Concatenative synthesis: Diphones (2)

target: \( w \text{ I n t r= d eI} \)

\[ \_w \text{ w-I I-n n-t t-r= r=-d d-eI eI-} \]

PSOLA pitch manipulation
Concatenative synthesis
Unit selection

target: \text{Which of these?}\
\text{Let's discuss the question of interchanges another day.}

acoustic unit database
units = (di-)phone segments recorded in natural sentences (natural intonation)
Basic idea of unit selection synthesis

- record naturally spoken utterances
  - with natural prosody
  - full sentences
- record many variants of each sound
  - usually, 1-10 hours of speech by the same speaker
- for a given target sentence, select the chain of units which “fit best”
  - crucial to define suitable measure of “fitting”
- do no (or only very little) signal processing on concatenated audio
  - because that would degrade the quality
Steps in run-time unit selection

Target prediction

- NLP-based prediction of synthetic target utterance
  - chain of phoneme symbols
  - sentence mode
  - stressed syllables
  - accents and boundaries

- Rich linguistic information represented, e.g., as a tree structure inside the system
Steps in run-time unit selection

Target prediction (2)

- Convert phoneme chain into a unit target chain
- Question of unit size
  - phone units
  - diphone units
  - halfphone units
- For each target, compute a “target feature vector” from the rich linguistic information
  - the target feature vector represents the “ideal” unit for the utterance we want to synthesize
Steps in run-time unit selection

Target prediction (3): Example

show example here
Steps in run-time unit selection
Pre-selection of candidates

First idea: get all possible candidates for each target
- e.g., for the target “a:”, get all [a:] units from database
- disadvantage: slow – too many candidates! (e.g., 3000 [a:] in corpus)

Better idea: get a preselection of “probably suitable” candidates
- walk through a “Classification and Regression Tree” (CART) using the target feature vector
Steps in run-time unit selection

Pre-selection of candidates (2): CART

- decision tree
- each decision node tests one of the target features
- stop at leaf or when desired number of candidates is reached
- only part of the target feature vector taken into account

draw CART here
Steps in run-time unit selection

Target costs and join costs

- two main measures of “good fit”
  - target costs: suitability of unit for target
  - join costs: acoustic continuity of two adjacent units

- target costs computed by comparing the target feature vector with each unit feature vector

- join costs computed by calculating the acoustic continuity between all pairs of candidates for adjacent targets
Steps in run-time unit selection
Dynamic programming

Viterbi algorithm: an efficient way of finding the optimal path through the unit candidates
- minimise combination of target and join costs

basic idea: be efficient by discarding paths that “cannot win” early in the process
- compute costs for partial paths;
- for each node, only the best (=lowest cost) of the partial paths leading to that node is retained
- iteratively apply that method from left to right => choosing the best partial path is always a local decision
Steps in run-time unit selection
Dynamic programming

“synthesize” = [s l n T @ s al z]
"synthesize" = [s l n T @ s a l z]

Steps in run-time unit selection
Dynamic programming

Partial path costs:
53.1
17.3
“synthesize” = [s l n T @ s al z]

Partial path costs:
53.1 - 3.1 - 2.2 = 50.4
17.3 + 8.5 + 2.2 = 28.0
Steps in run-time unit selection

Concatenation of audio segments

- “glue audio snippets together”
- simplest case: take audio as is
  - usually, try to find suitable concatenation points, for example at boundary between pitch periods
- more sophisticated: overlap-add
  - “fade out one unit – fade in the other”
  - to reduce discontinuities
- optionally, pitch smoothing / duration adaptation
  - same technology as used in diphone synthesis
  - but as little of it as possible
Creating a unit selection voice

Coverage

Case 1: general domain

- need to be able to say everything in the language
- “optimal” coverage = theoretical concept, not practically attainable: have every diphone in every possible prosodic realisation
- “minimal” coverage = every diphone in the language at least once
- but in practice, need also prosodic variation => the frequent diphones should occur in several prosodic realisations (position in sentence, accent, ...)
- also, many of the diphones are rare => create them from suitable half-phones
Creating a unit selection voice

Coverage

Case 2: limited domain
- need to be able to say things in a limited domain only
- e.g., telling the time, weather forecasts, ...
- can attain "optimal" coverage

Case 3: domain-oriented but open
- strive for "optimal" coverage within the domain, and at least "minimal" coverage outside it
- e.g., a speaking mp3 player (needs to speak arbitrary song titles), a flight-booking system (needs to speak names of many destinations)
Creating a unit selection voice

Recording

- put a trained actor in front of a microphone
- watch that pronunciation is close to intended pronunciation
- watch that voice quality stays constant (to the extent possible)
- recording time = quite some hours, usually on several days
- ideally, record individual sentences directly as individual files onto hard disk => save manual segmentation time
Creating a unit selection voice

Labelling

- mark phone boundaries in the audio signal
  - to know the start+end time of units
- hand-labelling = extremely time-consuming = expensive
- automatic labelling
  - use speech recognition technology
  - e.g. forced alignment: predict phoneme chain using TTS, and force the HMMs to find it in the audio signal
  - manual inspection needed to find serious errors