Decision Tree Usage for Incremental Parametric Speech Synthesis

Timo Baumann
baumann@informatik.uni-hamburg.de
Decision Tree Usage for Incremental Parametric Speech Synthesis
Decision Tree Usage for Incremental Parametric Speech Synthesis
Decision Tree Usage for **Incremental** Parametric Speech Synthesis

- ability to change ongoing speech output
- needed to cope with unexpected events in interactive use-cases
Speech Output in Typical Systems

There's an appointment today at 8:30 titled: 'presentation' with the note: 'do not miss'.

- conventionally: generate, synthesize and deliver utterance as a whole
Speech Output in Typical Systems

- potentially slow, as all processing is utterance-initial
  → reason for canned speech in deployed dialogue systems

There's an appointment today at 8:30 titled: 'presentation' with the note: 'do not miss'.
Speech Output in Typical Systems

There's an appointment today at 8:30 titled: 'presentation' with the note: 'do not miss'.

- inflexible: unable to change the ongoing utterance (neither the content nor the delivery parameters)
  - no way to react to the listener or the environment
Speech Output in Typical Systems

- inflexible: unable to change the ongoing utterance (neither the content nor the delivery parameters)
  - no way to react to the listener or the environment

There's an appointment today at 8:30 titled: 'presentation' with the note: 'do not miss'.

- user feedback
- when?
- crossmodal interaction
- noise
- calendar entry changes
Potentially Better: Incremental Speech Output

- There's an appointment today at 8:30 titled: 'presentation' with the note: 'do not miss'.

- generate, synthesize and deliver the utterance in small chunks
  - smaller chunks, higher flexibility
  - but (re)compute with as much context as is available or needed
Potentially Better: Incremental Speech Output

- generate, synthesize and deliver the utterance in small *chunks*

  → smaller chunks, higher flexibility
  → but (re)compute with as much context as is available or needed

There's an appointment today at 8:30 titled: 'presentation' with the note: 'do not miss'.
Potentially Better: Incremental Speech Output

current point in time

There's an appointment today at 8:30 titled: 'presentation' with the note: 'do not miss'.

- less utterance-initial processing ➔ faster onset
Potentially Better: Incremental Speech Output

There's an appointment today at 8:30 titled: 'presentation' with the note: 'do not miss'.

- incremental output may take changes into account
- react and adapt to user feedback / requests / noise

When?

we tackled noise in information presentation in Buschmeier et al., SigDial 2012, we dealt with cars going driving around in SigDial 2013, Dialog-in-Motion Workshop 2014.
what I'm trying to say is:

incremental speech synthesis is a requirement for highly responsive behaviour in interactive systems
“Just-in-Time“ Incremental Speech Synthesis

„Your flight in May, to Florence, has been confirmed by the airline."

not?

full utterance

phrases

words

syllables

phones

HMM states w/ durations

feature vectors

speech audio

as implemented in InproTK (Baumann&Schlangen SDCTD 2012)
„Just-in-Time“ Incremental Speech Synthesis

„Your flight in May, to Florence, has been confirmed by the airline.“

full utterance

phrases
words
syllables
phones
HMM states w/ durations

feature vectors

speech audio

as implemented in InproTK (Baumann&Schlangen SDCTD 2012)
„Just-in-Time“ Incremental Speech Synthesis

„Your flight in May, to Florence, has been confirmed by the airline.“

full utterance
phrase_1 phrase_2 ...

phrases
words
syllables
phones
HMM states w/ durations
feature vectors
speech audio

as implemented in InproTK (Baumann&Schlangen SDCTD 2012)
"Just-in-Time" Incremental Speech Synthesis

"Your flight in May, to Florence, has been confirmed by the airline."

- full utterance
- phrase 1
- phrase 2
- ...
- words: your, flight, in, May
- syllables
- phones
- HMM states w/ durations
- feature vectors
- speech audio

as implemented in InproTK (Baumann & Schlangen SDCTD 2012)
„Just-in-Time“ Incremental Speech Synthesis

„Your flight in May, to Florence, has been confirmed by the airline.“

as implemented in InproTK (Baumann&Schlangen SDCTD 2012)
“Just-in-Time” Incremental Speech Synthesis

Your flight in May, to Florence, has been confirmed by the airline.

Full utterance

- Phrases
  - Phrase 1
  - Phrase 2
  - ...

- Words
  - Your
  - Flight
  - In
  - May
  - ...

- Syllables
  - *
  - *
  - *
  - ...

- Phones
  - j u r f l a r i t i
  - ...

HMM States with Durations

Feature Vectors

Speech Audio

As implemented in InproTK (Baumann & Schlangen SDCTD 2012)
“Just-in-Time” Incremental Speech Synthesis

„Your flight in May, to Florence, has been confirmed by the airline.“

as implemented in InproTK (Baumann&Schlangen SDCTD 2012)
„Just-in-Time“ Incremental Speech Synthesis

„Your flight in May, to Florence, has been confirmed by the airline.“

as implemented in InproTK (Baumann&Schlangen SDCTD 2012)
“Just-in-Time” Incremental Speech Synthesis

„Your flight in May, to Florence, has been confirmed by the airline.“

as implemented in InproTK (Baumann & Schlangen SDCTD 2012)
"Just-in-Time" Incremental Speech Synthesis

"Your flight in May, to Florence, has been confirmed by the airline."

as implemented in InproTK (Baumann&Schlangen SDCTD 2012)
„Your flight in May, to Florence, has been confirmed by the airline.“

„Just-in-Time“ Incremental Speech Synthesis

... vocode just enough to keep soundcard buffers full
"Just-in-Time" Incremental Speech Synthesis

"Your flight in May, to Florence, has been confirmed by the airline."

| Phrases | \( \text{phrase}_1 \) | \( \text{phrase}_2 \) |
|---------|------------------------|
| Words   | your | flight | in | May | ...
| Syllables | * | | | *+ |
| Phones  | Jur | fl | ar | ti |
| HMM states w/ durations |
| Feature vectors |
| Speech audio |

... Dutoit et al., 2011: emission prob. estimation can be performed stepwise, with just a few future HMM states known (i.e., works locally)
Your flight in May, to Florence, has been confirmed by the airline.

... one full phrase of lookahead gives near perfect prosody (Baumann & Schlangen 2012)

... Dutoit et al., 2011: emission prob. estimation can be performed stepwise, with just a few future HMM states known (i.e., works locally)
"Just-in-Time“ Incremental Speech Synthesis

"Your flight in May, to Florence, has been confirmed by the airline."

... one full phrase of lookahead gives near perfect prosody (Baumann & Schlangen 2012)

... link linguistic/symbolic processing with parametric processing using decision trees

... Dutoit et al., 2011: emission prob. estimation can be performed stepwise, with just a few future HMM states known (i.e., works locally)
„Just-in-Time“ Incremental Speech Synthesis

„Your flight in May, to Florence, has been confirmed by the airline.“

... one full phrase of lookahead gives near perfect prosody (Baumann & Schlangen 2012)

... link linguistic/symbolic processing with parametric processing using decision trees

... Dutoit et al., 2011: emission prob. estimation can be performed stepwise, with just a few future HMM states known (i.e., works locally)

- goal: better incremental HMM state selection
  - without re-training synthesis voices
  - without sacrificing non-incremental performance
HMM State Selection

- most TTS systems (such as MaryTTS) use decision trees
  - separate trees for MCEP, STR, f0 streams, and state duration
- feature sets using various types of information
  - MaryTTS: roughly 100 features

- many features are *non-local*
  such as „how many phonemes until end of utterance?“

→ non-local features
  are not available in incremental processing
Classification of features among two dimensions as implemented in InproTK (Baumann & Schlangen SDCTD 2012)

„Your flight in May, to Florence, has been confirmed by the airline.“

- level of linguistic abstraction
  - higher-level information spans longer time frames

...
Classification of features among two dimensions

„Your flight in May, to Florence, has been confirmed by the airline."

- temporal direction: past, present, future
  - past is generally available, future requires lookahead

as implemented in InproTK (Baumann&Schlangen SDCTD 2012)
<table>
<thead>
<tr>
<th>Feature</th>
<th>past</th>
<th>current</th>
<th>future</th>
</tr>
</thead>
<tbody>
<tr>
<td>full sentence</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>phrase/accentuation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>word</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>syllable</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>phone</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
A Classification of Features: MaryTTS feature counts for German

<table>
<thead>
<tr>
<th>Feature</th>
<th>past</th>
<th>current</th>
<th>future</th>
</tr>
</thead>
<tbody>
<tr>
<td>full sentence</td>
<td>—</td>
<td>5</td>
<td>—</td>
</tr>
<tr>
<td>phrase/accentuation</td>
<td>11</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>word</td>
<td>2</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>syllable</td>
<td>3</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>phone</td>
<td>20</td>
<td>10</td>
<td>19</td>
</tr>
</tbody>
</table>
A Classification of Features: MaryTTS feature counts for German

<table>
<thead>
<tr>
<th></th>
<th>past</th>
<th>current</th>
<th>future</th>
</tr>
</thead>
<tbody>
<tr>
<td>full sentence</td>
<td>—</td>
<td>5</td>
<td>—</td>
</tr>
<tr>
<td>phrase/accentuation</td>
<td>11</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>word</td>
<td>2</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>syllable</td>
<td>3</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>phone</td>
<td>20</td>
<td>10</td>
<td>19</td>
</tr>
</tbody>
</table>

- generalize features into classes that represent lookahead requirements in an incremental system
## A Classification of Features: MaryTTS feature counts for German

<table>
<thead>
<tr>
<th></th>
<th>past</th>
<th>current</th>
<th>future</th>
</tr>
</thead>
<tbody>
<tr>
<td>full sentence</td>
<td>—</td>
<td></td>
<td></td>
</tr>
<tr>
<td>phrase/accentuation</td>
<td>11</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>word</td>
<td>2</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>syllable</td>
<td>3</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>phone</td>
<td>20</td>
<td>10</td>
<td>19</td>
</tr>
</tbody>
</table>

- generalize features into classes that represent lookahead requirements in an incremental system
Experiment: What if a feature class is not available?

- substitute a default feature value
- training:
  - synthesized 600 utterances and recorded all feature usages (and their values) in the decision trees
  - determine default feature values
    - most common value for categorial features
    - mean value for numeric features
- test:
  - re-process, substituting features of a class by their defaults
- measure (numeric) deterioration of resulting HMM states
  (z-normalized mean absolute error)
Results per Feature Class

sharp drop for spectral / excitation features
(already very good with ~2 phones of future context)
f0 and duration level off only slowly
(→need more context)
Conclusion

- decision tree features can be missing during incremental processing
  - substitute with default values

- classified features into classes which are meant to correspond to context/lookahead requirements

- the more context, the better the results
  - relatively small lookahead (1 syllable/2 phones) enough for voice quality (MCEP and STR)
  - prosody (duration and f0) in contrast, requires a large lookahead, or more advanced methods

- there's no simple good or bad, but a continuous improvement the more context is available
Thank you.

baumann@informatik.uni-hamburg.de,
get the code at inprotk.sf.net.

Funded by a Daimler and Benz Foundation PostDoc grant.

Thanks to Sven Mutzl, who performed some initial analyses while on an internship with Prof. Kai Yu at Yiao Tong University, Shanghai, China.
Raum für Notizen
Default feature values vs. „properly“ dealing with missing features