Predicting the Micro-Timing of User Input for an Incremental Spoken Dialogue System that Completes a User's Ongoing Turn

Timo Baumann
and
David Schlangen

mail@timobaumann.de

http://www.ling.uni-potsdam.de/~timo

Incremental SDSes have come a long way

- *incremental* ≙ processing during user input

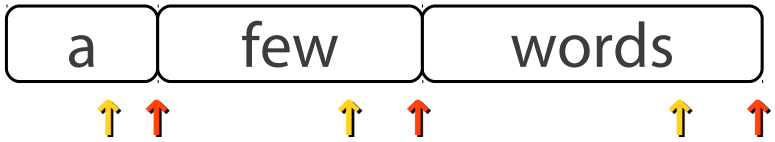
- incremental NLU during user input:
  - produce partial semantic representations
  - often able to grasp the full utterance before it is over
  - able to decide the point of maximum understanding

we can try to *generate completions*
of what the user has started to say

(Atterer et al., 2009, Sagae et al., 2009; Heintze et al., 2010; DeVault et al., 2009)

Incremental ASR is very fast

- when does the ASR *hear* words?
  - first intuition around ¾ of the word
  - final recognition around end of the word



we can predict words *before they are over*,
can we also predict *how much is missing*?

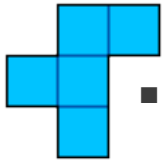
(Baumann et al., 2009)

Predicting the Micro-Timing of User Input for an Incremental Spoken Dialogue System that Completes a User's Ongoing Turn

- I've just described what I mean by Micro-Timing
- you know Incremental Spoken Dialogue Systems
- I'll show off our Micro-Timing capabilities by synchronously completing ongoing turns
  - this is not a terribly important capability, rather a good technology demonstration – if we can do this, you can probably do a lot of other things with the timing model

Why we really want to monitor the user's micro-timing

1. to help the user when he's struggling for words



- „take the uh … piece … that looks like **an F?“**
- rather a split utterance not a co-completion

2. micro-timing for back-channels and next turns
   - most likely in combination with prosodic cues
3. use it to monitor the user's *fluency*:
   - by measuring the amount of unexpected deviation

- nonetheless, let's stick with co-completions …

(Schlangen et al., 2009)

Co-Completing the User

- computers should certainly not *always complete* a turn that they understand (not even often)
- however, this can be an efficient interactional device if used occasionally in certain situations
  - conversational systems, negotiation training, …
- frequency of occurrence in human dialogue:
  - sentence cooperations in task-oriented German: 3.4 %
  - split utterance boundaries in the BNC: 2.8 %

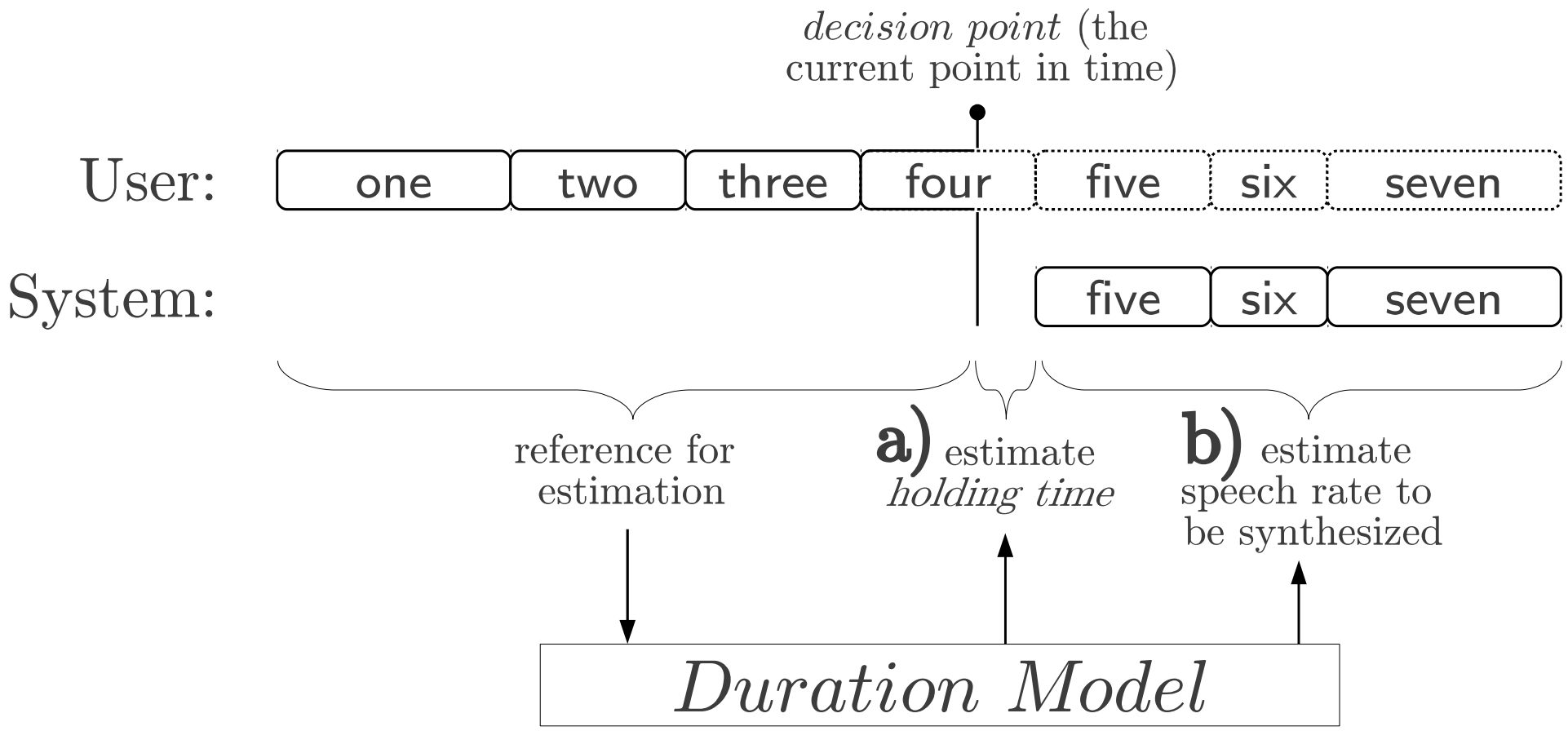
(Skuplik 1999; Purver et al., 2009)

Predicting the Micro-Timing of a User's Ongoing Words

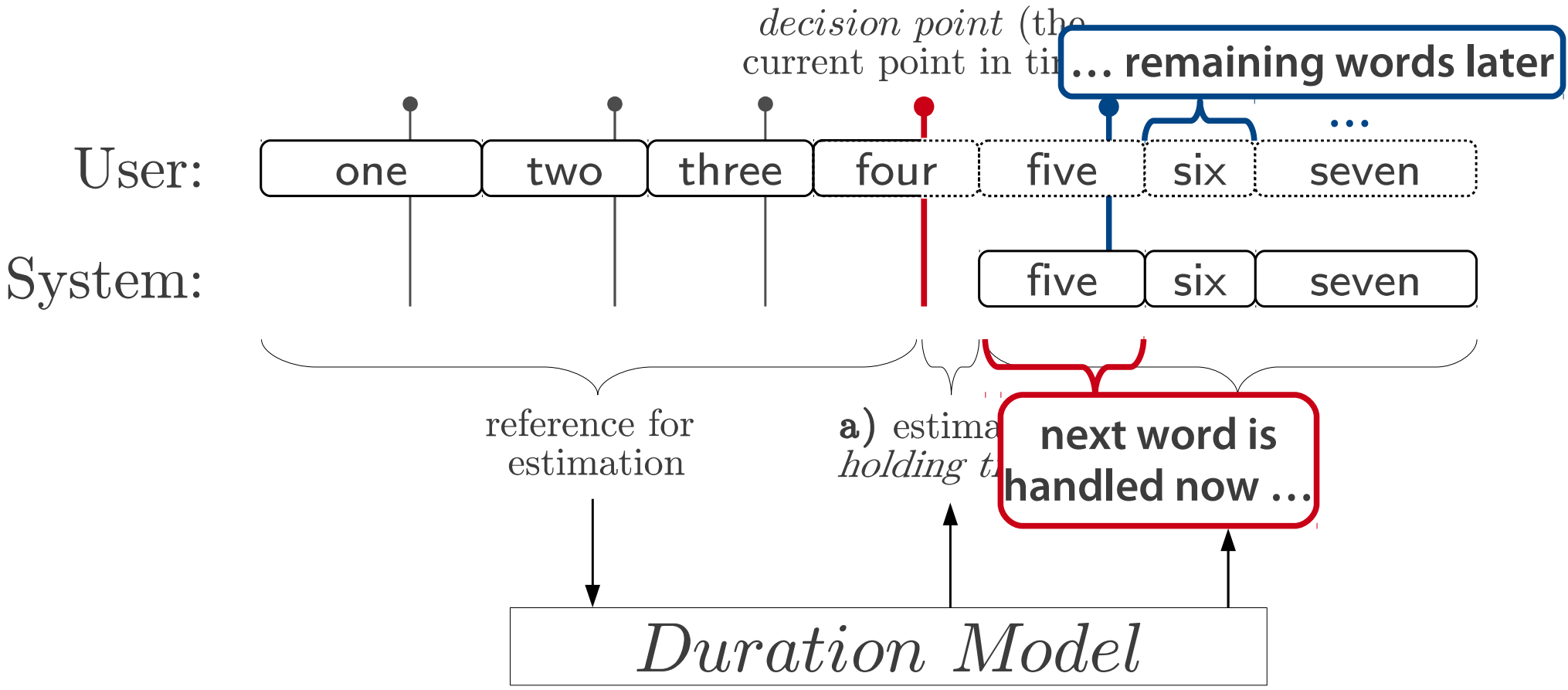
Our Example Task:

- let's *shadow* the user while she is speaking, i.e. say the same thing that she says and in the same way
  - we assume that she's *reading* a text *that we know*
  - identical to *synchronous reading* task (Cummins 2002)
- to be able to shadow we have to
  - identify the user's current word before it's over
  - estimate the time remaining for the current word
  - estimate the speech rate for the next word

The Task

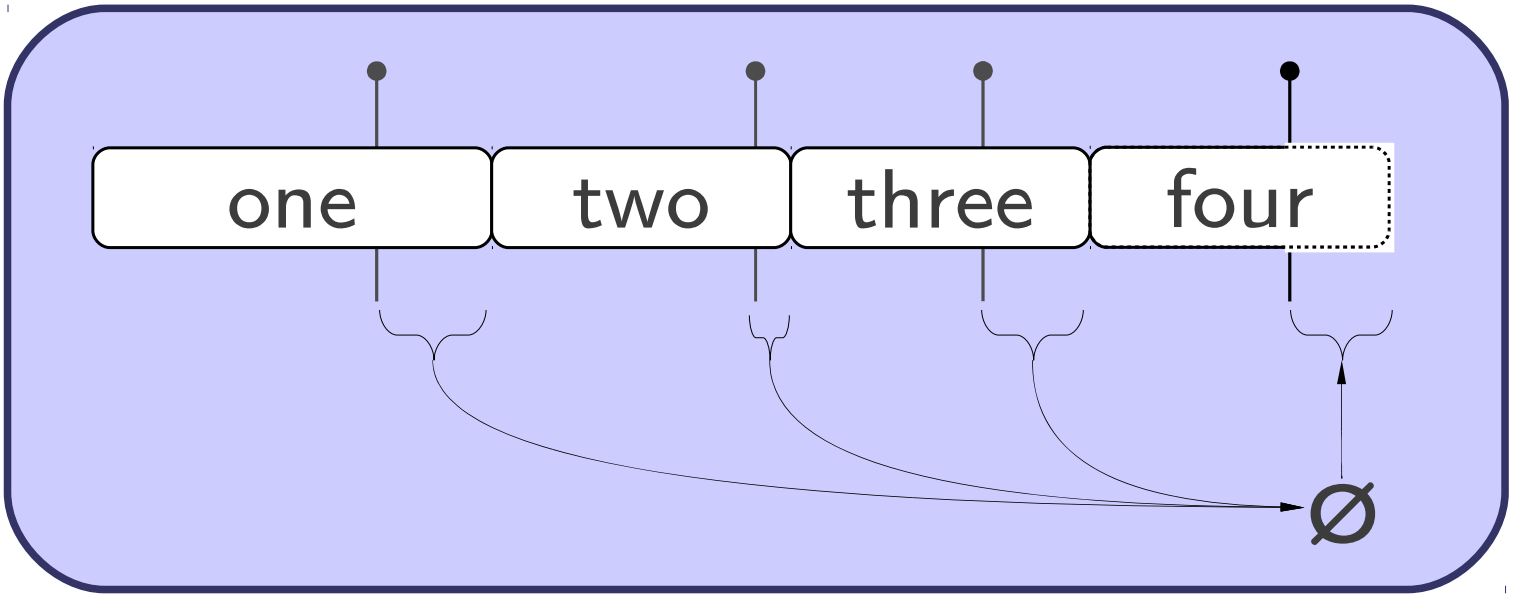

decision point (the current point in time)
User:
one
two
three
four
five
six
seven
System:
five
six
seven
reference for estimation
a) estimate *holding time*
b) estimate speech rate to be synthesized
Duration Model

Shadowing iteratively word-by-word



Strategy 1: average ASR lookahead (baseline)

- assume that ASR results are similarly timed
- use average lookahead as the remaining time



- solves task (**a**), but doesn't give tempo for next word

We need a real duration model

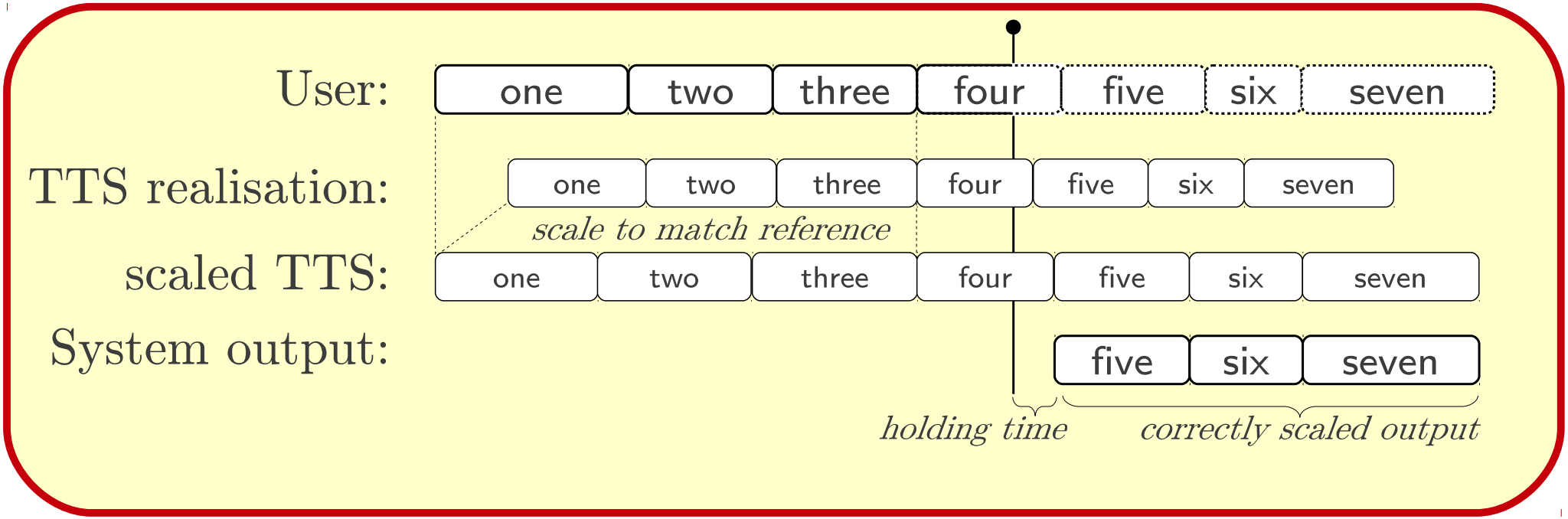
- given some partial input (words and durations)
- and the expected completion (words, no durations)
- assign expected durations for the completion

- Start from “canonical durations” for the input
  - compare expected and actual tempo
  - (linearly) scale canonical durations to match the user
- What model can generate the canonical durations?
  - hey, TTSs have very good duration models!

Strategy 2: Analysis-by-Synthesis

- listen to what is being said (prev.words, curr.w.), predict what will be said (compl.),
- feed combined full utterance to (symbolic) TTS
- $scaling\ factor := \frac{\text{length}_{User}(prev.words)}{\text{length}_{TTS}(prev.words)}$
- $holding\ time := \text{length}_{TTS}(curr.w) * scaling\ factor - \text{length}_{User}(curr.w)$
- scale completion with scaling factor, send to (acoustic) TTS
- play output at predicted time

Strategy 2: Analysis-by-Synthesis



Experiment Setup

- recognize utterances from a known corpus („Nordwind und Sonne“ – *Kiel Corpus of Read Sp.*)
- for every word:
  - how long before its end do we recognize it?
    - only if we’re before the end, can we act on time
  - predict how much time is remaining *(holding time)*
  - predict the duration of the next word
  - demo: talk *in sync* with the speaker

Results

- words are recognized sufficiently early (μ = -134 ms)
- *holding time* prediction and next words' durations are significantly improved by Analysis-by-Synthesis (std dev = 85 ms / 94 ms)
- median absolute error (MAE = 74 ms) is similar to human performance for *synchronous speech* (56 ms)

… alright, but *how does it sound?*

(Cummins 2002)

How does it sound?

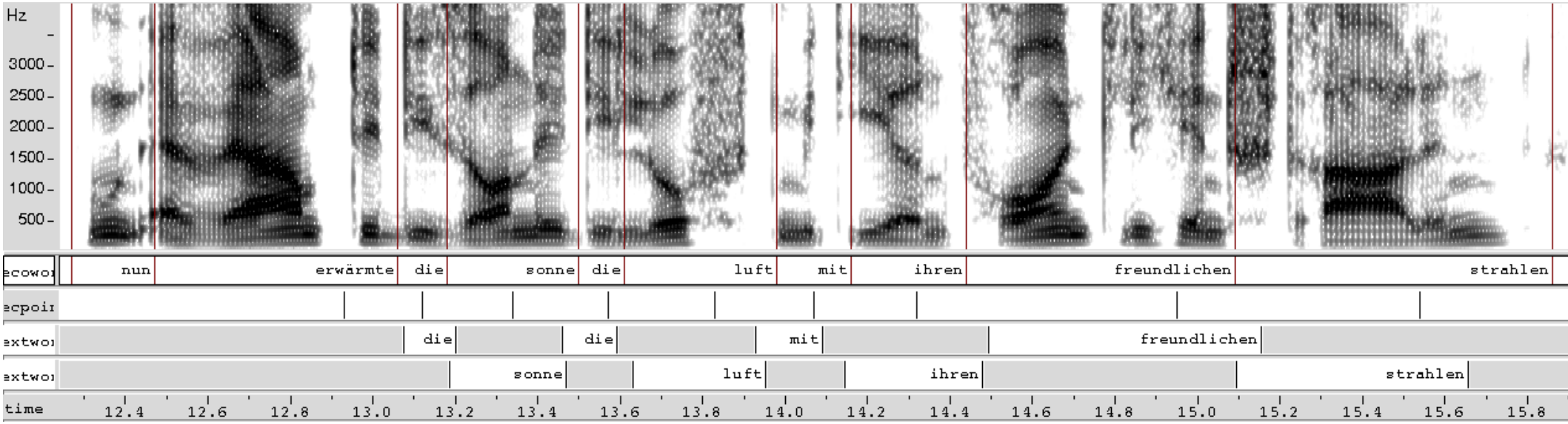
Excerpt from “The Northwind and the Sun”

Endlich gab der Nordwind den Kampf auf.
Nun erwärmte die Sonne die Luft mit ihren freundlichen Strahlen
und schon nach wenigen Augenblicken zog der Wanderer seinen Mantel aus.

At last the North Wind gave up the attempt.
Then the Sun shined out warmly,
and immediately the traveler took off his cloak.

(IPA 1999; drawing by Milo Winter, 1919, public domain)

How does it sound?



open audio …

open video …

open audio …

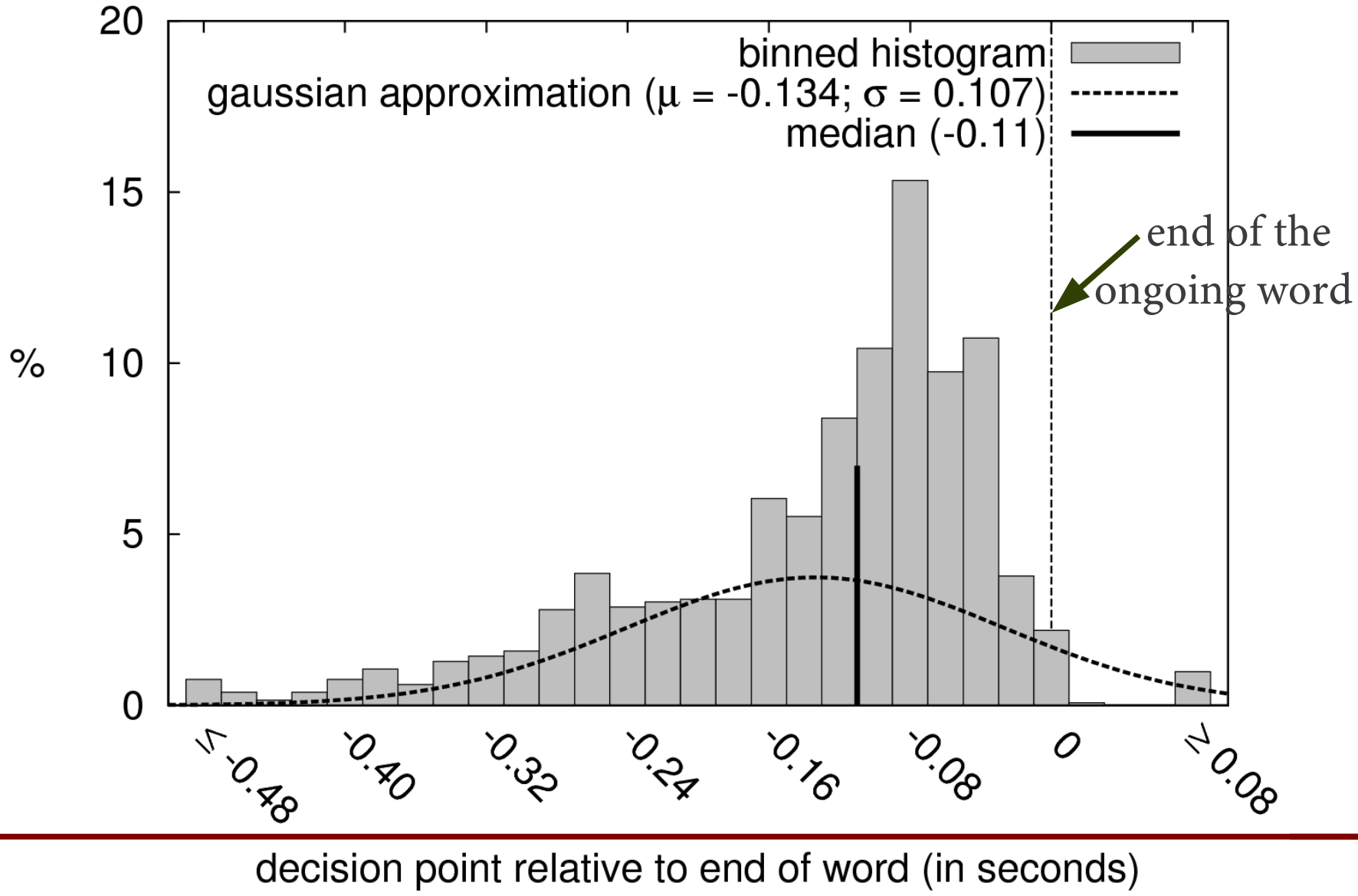
Thank you !

Questions or Comments ?

mail@timobaumann.de

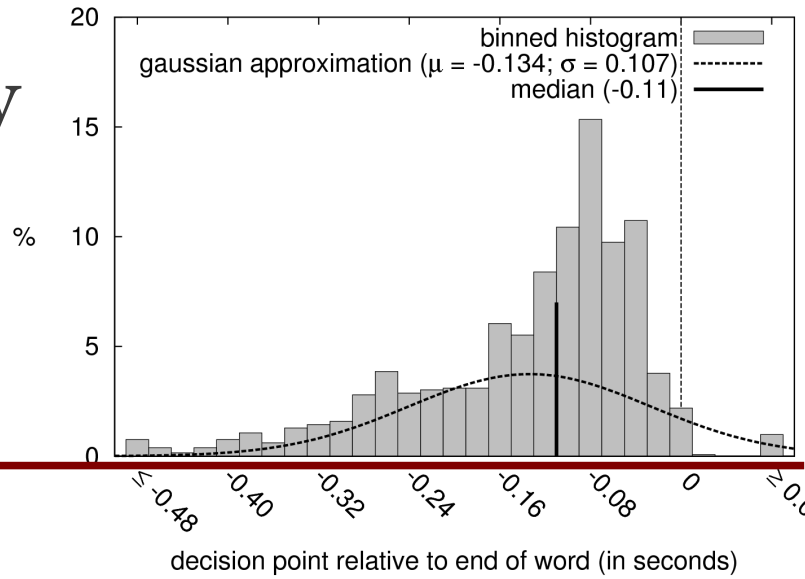
http://www.ling.uni-potsdam.de/~timo/pub/shadowing/

Results: raw ASR timing



Results: raw ASR timing

- recognition happens before the word's end (good!)
- often there is still plenty of time (μ=134ms)
  - processing, TTS synthesis, soundcard delays, …
  - we could use this time for ASR to become stable
- high between-speaker variability (μ between 97 and 237 ms)



Results: Holding-Time Estimation

model	bias		jitter	
	mean	median	std dev	MAE
baseline: all	-134	-110	107	110
baseline $-\mu$	0	23	107	63
ASR-based: all	-2	19	105	60
IPU-internal	26	33	82	51
IPU-final	-148	-143	87	142
TTS-based: all	-3	4	85	45
IPU-internal	12	11	77	41
IPU-final	-78	-76	83	79

Results: Holding-Time Estimation

- both strategies reduce bias close to zero
  - when distinguishing between IPU-internal and IPU-final words, TTS-based strategy is significantly better
- TTS-based strategy significantly reduces jitter
  - median absolute error (MAE) similar to human performance for *synchronous speech* (Cummins 2002)
- IPU-internal and IPU-final predictions differ
  - likely due to final lengthening

Results: The Next Word's Duration

task	error distribution metric (in ms)			
	mean	median	std dev	MAE
TTS-based : duration	-5	4	75	45
+ **ASR-based** : onset	26	33	82	51
= end of word	25	30	100	81
+ **TTS-based** : onset	12	11	77	41
= end of word	7	10	94	74

Results: The Next Word's Duration

- duration prediction (with TTS-strategy) performs almost as good as in step 2
- however, the errors of step 2 and step 3 add up:
  - $\sigma_{\text{step 2 = 77 ms}} + \sigma_{\text{step 3 = 75 ms}} = 94$ ms
- deviation will likely increase for longer completions
  - underlines the need for an incremental TTS to allow instant adaptation of output as it occurs