

Incremental dialogue management

Wolfram Wingerath

23.01.2012



- ① Incremental dialogue management – basics
- ② Building an incremental dialogue manager
- ③ Evaluating incremental processors
- ④ References



Incremental dialogue management – basics

- 1 Incremental dialogue management – basics
 - Utterances in traditional dialogue systems
 - Sub-utterance phenomena
 - Numbers – an incremental dialogue system
 - Challenge: the revoke-commit-problem
- 2 Building an incremental dialogue manager
- 3 Evaluating incremental processors
- 4 References

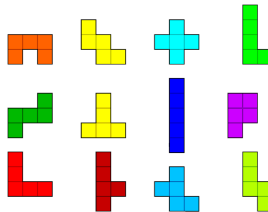


-

- natural dialogues proceed **continuously** and **incrementally**



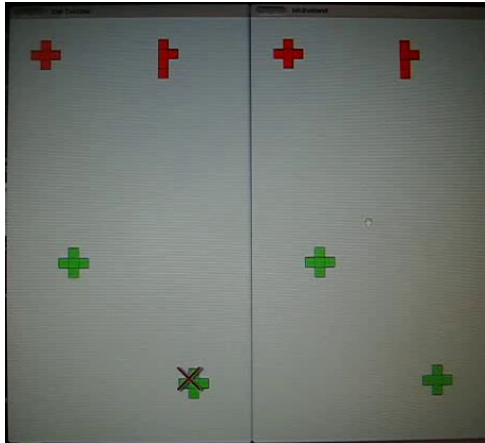
A very simple incremental dialogue manager



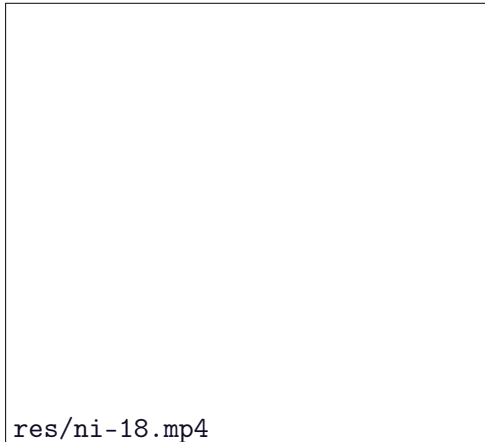
- an instructor tells the system which piece to take or to delete
- incremental vs. non-incremental version: In an overhearer evaluation, the **incremental version was preferred** w.r.t.
 - human-likeness
 - helpfulness
 - **reactivity**



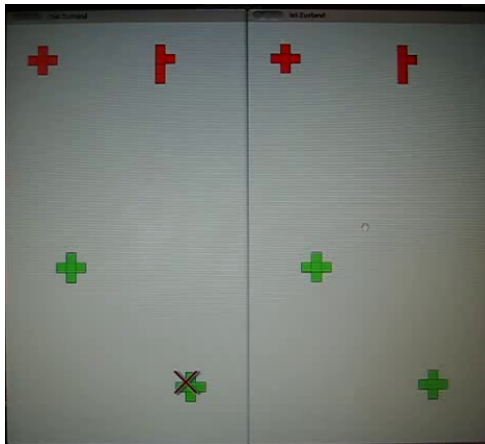
Demo: non-incremental version



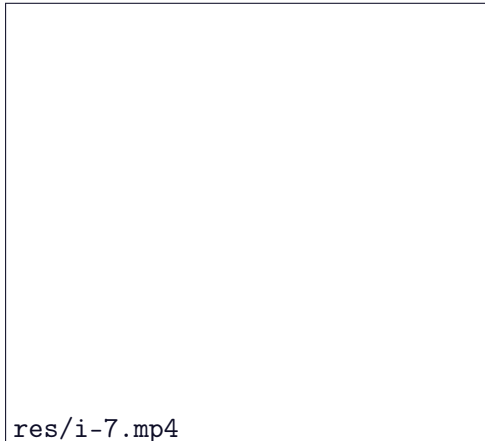
Demo: non-incremental version



Demo: incremental version



Demo: incremental version



Sub-utterance phenomena

A: From Boston uhm ... on Monday.

- dialogue partners normally also interact on **sub-utterance** level
- there are several sub-utterance phenomena
- a dialogue system that is able to 'understand' and produce them, is mostly perceived as more **natural** and more **efficient**



Hesitations

A: From Boston uhm ... on Monday.

- **unfilled pauses** are harder to recognise than **filled pauses** (e.g. 'erm'), because they may be confused with utterance ends
- a system might derive information from the fact that the user is hesitating or even offer help (→ **cooperative replies**)
- a system might also fill pauses to **appear more reactive**, e.g. start with a filled pause



Rapid turn-taking

A: From Boston.

B: Erm, hang on, I'll check.

- **seamless transitions** cannot be modelled with the typically used silence threshold-based utterance segmentation
- there are **other turn-taking cues** than silence, e.g.
 - variation of prosodic structure
 - pitch/loudness
 - syntax
 - semantics



Feedback utterances

A: From Boston on Monday.

B: ... uhu ...

- signal the **grounding status** without interrupting a turn
- can have a high influence on how the speaker continues (e.g. 'Huh?' vs. 'Mhm.')
- might even be implicitly requested



Interruptions

A: From Boston on...

B: Sorry, Boston airport is closed!

- **immediately address** a part of an utterance, e.g. when making a choice in a list of alternatives
- are useful, when user **notification** is required with high priority
- don't necessarily lead to a turn-change (e.g. short correction)



Relevant non-linguistic actions (RNLAs)

A: From Boston on Monday.



<http://de.wikipedia.org/wiki/Boston>

- better: **non-verbal actions**
- indicate degree of understanding (non-verbal feedback)

Relevant non-linguistic actions (RNLAs)

A: From Boston on Monday.



<http://de.wikipedia.org/wiki/Boston>

- better: **non-verbal actions**
- indicate degree of understanding (non-verbal feedback)

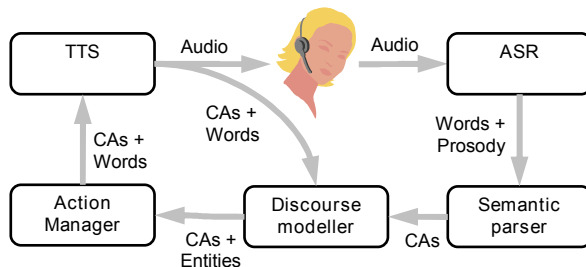
Recap: Why incremental dialog management?

hesitations	A: From Boston uhm ... on Monday.
rapid turn-taking	A: From Boston. B: Erm, hang on, I'll check.
feedback utterances	A: From Boston on Monday. B: ... uhu ...
interruptions	A: From Boston on... B: Sorry, Boston airport is closed!
relevant non-linguistic actions (RNLAs)	A: From Boston on Monday. B: [Boston lights up on a map]

- incremental dialogue system work on sub-utterances and thus might
 - give continuous feedback
 - react to feedback from the user while the system is speaking
 - → appear **more natural and efficient**

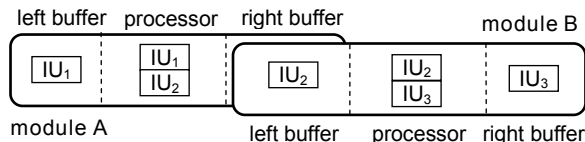


Numbers – an incremental dialogue system [SS09]



- turn-taking decisions are based on a combination of ASR, prosody and silence-thresholds
- latency of about 200ms (750ms for the non-incremental version)

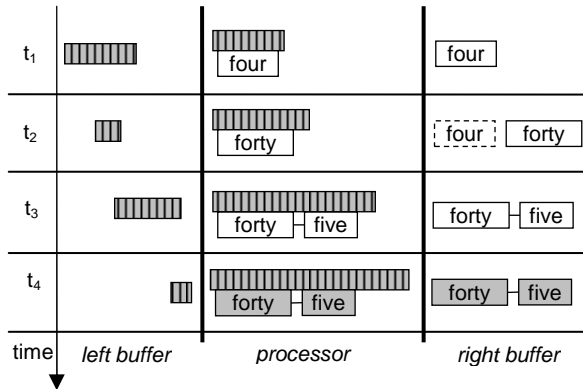
Connection between incremental processing modules



- **incremental units (IUs)**, i.e. data chunks, are processed from left to right
- output of one module is input of another module



Incremental processing in Numbers



- three different module operations: update, purge and commit



Demo: Numbers in action

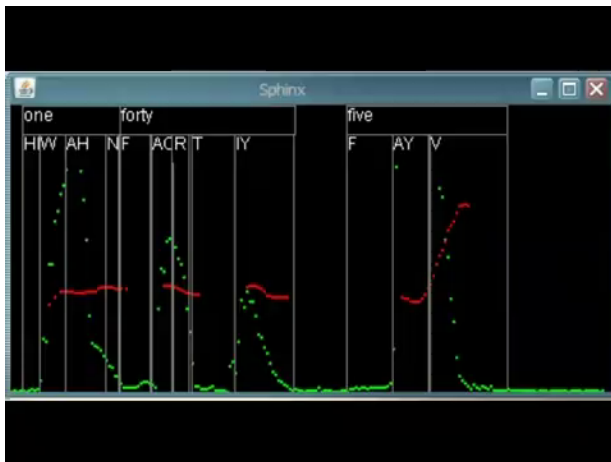


Demo: Numbers in action

`res/Numbers_spoken_dialogue_system_-_demo_use.mp4`



Demo: prosodic analysis in Numbers

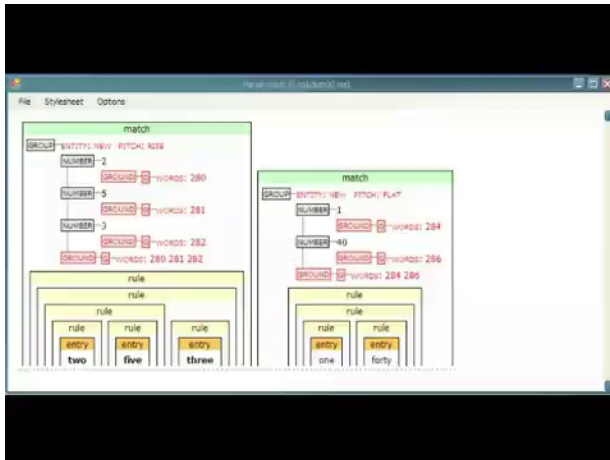


Demo: prosodic analysis in Numbers

res/Numbers_spoken_dialogue_system_-_demo_prosody.mp



Demo: incremental parsing in Numbers



Demo: incremental parsing in Numbers

[res/Numbers_spoken_dialogue_system_-_demo_parsing.mp](#)



Challenge: the revoke-commit-problem

- **unstable hypotheses**: Hypotheses may change as more of the utterance is heard (e.g. 'four' → 'fourty' → 'fourty-five')
- Problem:
What to do, if a committed hypothesis must be revoked?
 - internal state has to be updated (**covert repair**)
 - if erroneous output has been produced, the mistake has to be corrected (**overt repair**)



Solution 1: reducing hypothesis instability

- reduce instability of hypotheses by allowing more right context, i.e. by **sending hypotheses delayed**
 - the problem is not solved, it just occurs less frequent
 - responsiveness is reduced dramatically



Solution 2: ignoring the problem

- the system just produces the right output without further comment
 - the system has no record of the repair → **inconsistencies may arise**, when the user explicitly refers to them

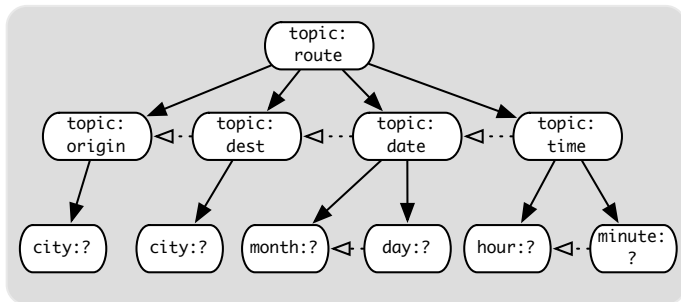


Solution 3: repair the damage

- 1 update internal state
- 2 check for erroneous output
- 3 initiate explicit repair



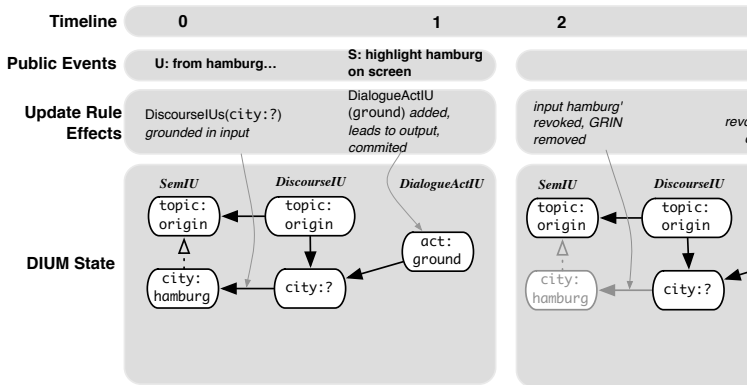
DIUM – a self-correcting incremental DM [BS11]



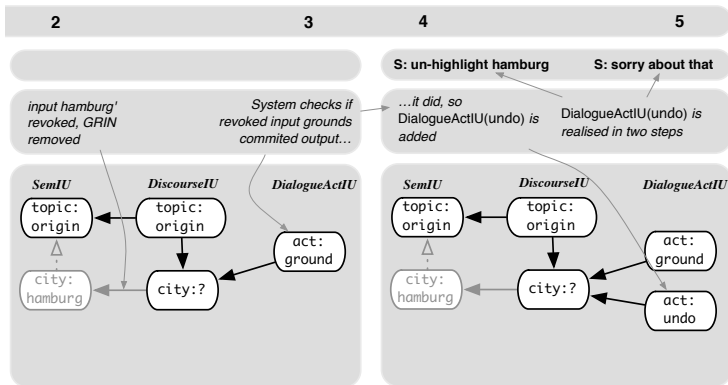
- DIUM stands for **IU**-based **D**ialogue **M**anager
- information states as graphs of IUs
- can correct committed hypothesis
- allows users to **over-answer** or switch topic



Repairing a committed hypothesis in DIUM I



Repairing a committed hypothesis in DIUM II



Building an incremental dialogue manager

- ① Incremental dialogue management – basics
- ② Building an incremental dialogue manager
 - Upgrade – adding a reactive layer
 - Rebuild – incrementalising the dialogue manager
 - Incremental speech generation in DEAL
- ③ Evaluating incremental processors
- ④ References



Upgrade – adding a reactive layer

- dialogue manager still works on **full utterances only**
- an additional component computes reactions from sub-utterance information
- attractive, because tried-and-tested traditional dialogue management paradigms can be used
- may improve the system, e.g. plausible feedback utterances can be produced



But...

A: Take the green block ...

B: Uhu.

A: ... and place it in the ...

B: Yeah?

A: ... middle of the board.

B: OK.

B: I'm sorry, what did you say?

- reactive layer and main dialogue manager can get **out of sync**
- responsiveness remains low

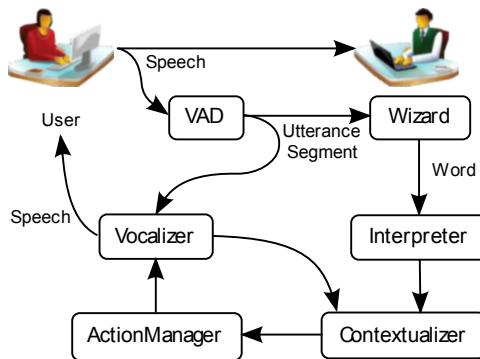


Rebuild – incrementalising the dialogue manager

- a **new context representation** is needed that
 - can be updated with partial information
 - tracks grounding state

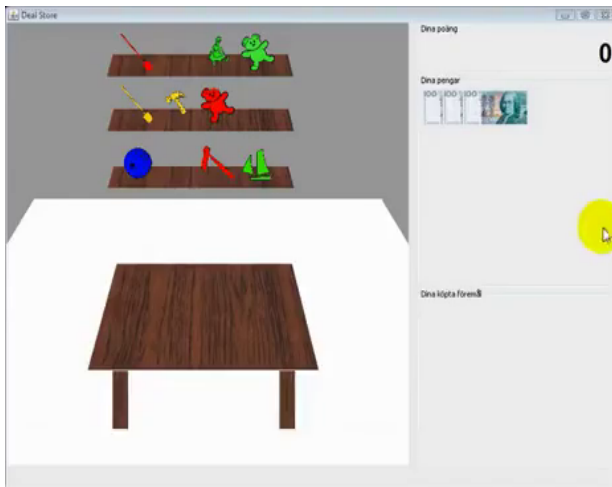


Incremental speech generation in DEAL

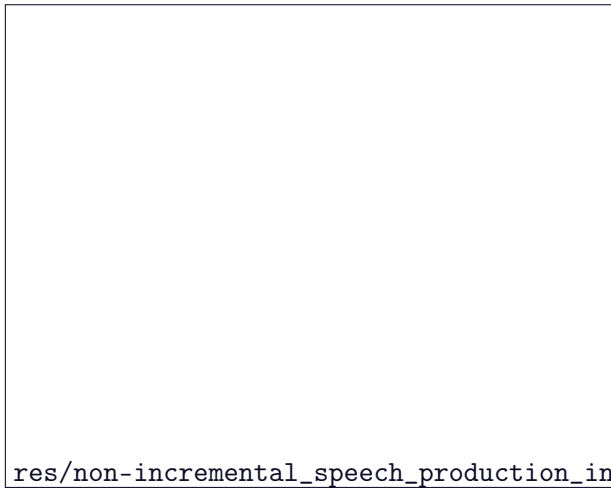


- system not fully functional yet → **wizard** as replacement for automatic speech recognition (ASR)

Demo: non-incremental version



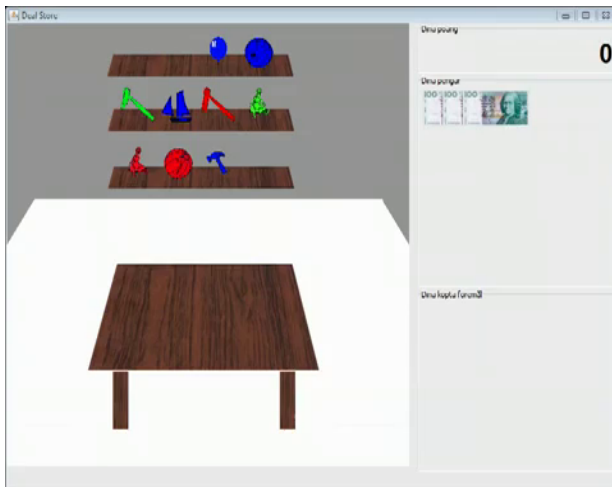
Demo: non-incremental version



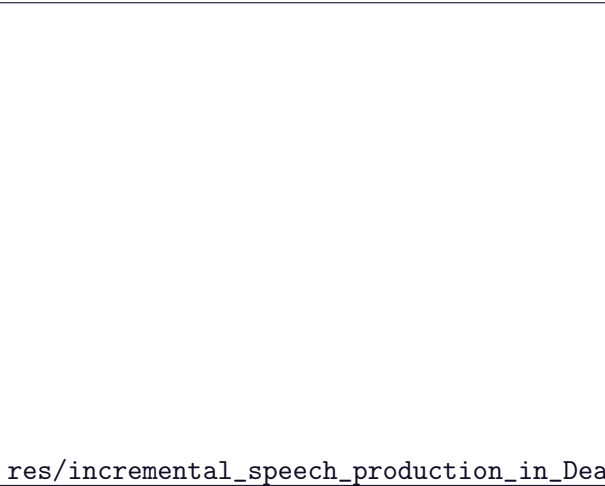
res/non-incremental_speech_production_in_Deal_-_demo



Demo: incremental version



Demo: incremental version



Evaluating incremental processors

- 1 Incremental dialogue management – basics
- 2 Building an incremental dialogue manager
- 3 Evaluating incremental processors**
 - Two kinds of gold standards
 - Metrics
 - Optimising incremental processors
- 4 References



Evaluating incremental processors [BBS11]

- incremental processors
 - generate partial results given partial input
 - are strongly interconnected
- evaluating a given processor means not only comparing, but **measuring the similarity** between its actual and its ideal output (gold standard)

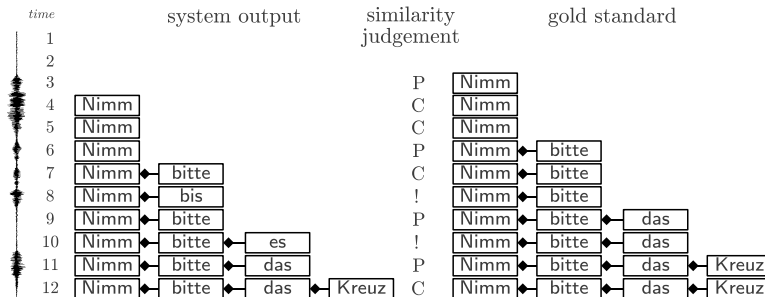


Two kinds of gold standards

- the questions are:
 - **What** is the ideal output for a given input?
 - **How** can it be generated?

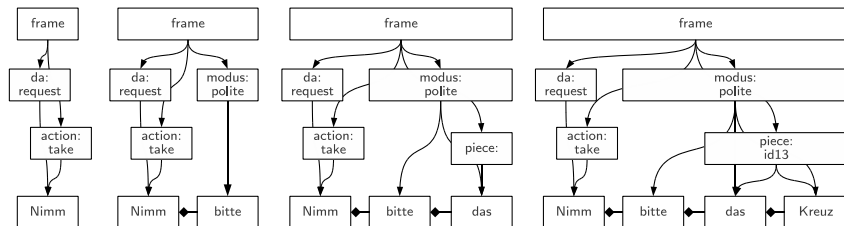


Incremental gold standards



- useful when available
- often available for processors like ASR

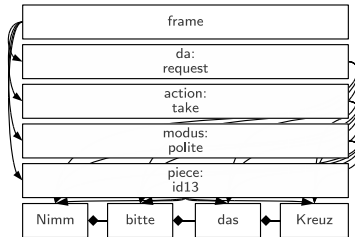
Non-incremental gold standards I



- it is often hard to find an **alignment** of ideal output and given input
- example: typically, only the final correct semantics is available

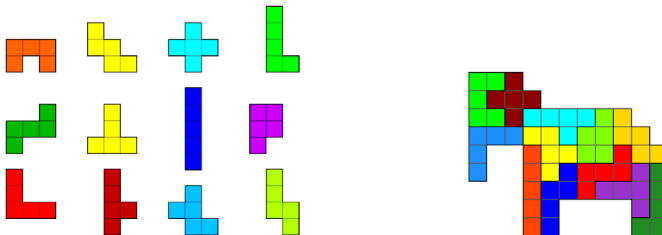


Non-incremental gold standards II



- unreasonable assumption: complete input is known from the start
- gradual **approach** towards the **gold standard** can be measured
- timeliness cannot be measured well

The Pentomino puzzle domain



- an instructor tells a follower (or the wizard) how to place the pieces

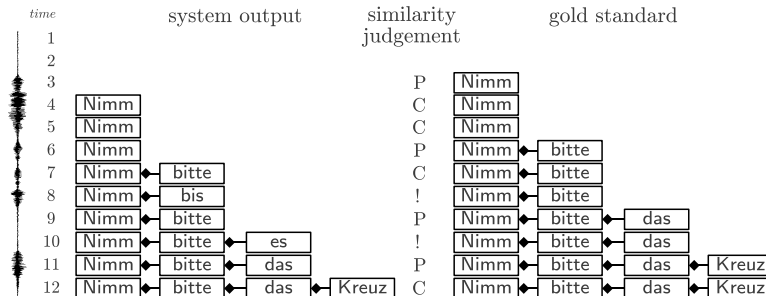


Metrics

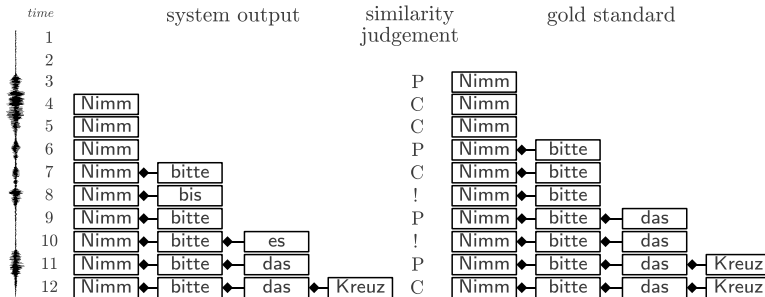
- three dimensions for output evaluation:
 - **content**
 - **timing**
 - **evolution** of incremental results



Similarity metrics

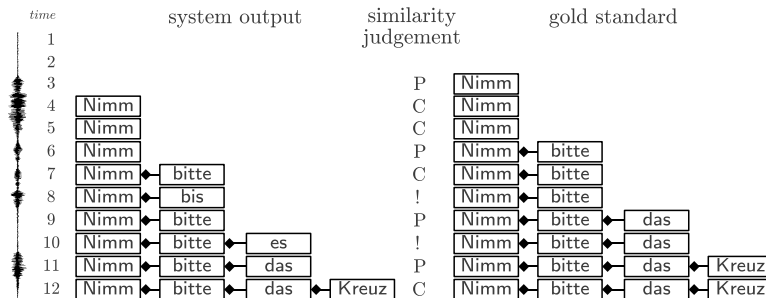


- measure similarity of output w.r.t. a gold standard
 - (delay-discounted) **correctness**: proportion of ideal intermediate results
 - **p-correctness**: proportion of results that are a prefix of their gold standard



- measure timing of output w.r.t. a gold standard
 - **FO**: first occurrence of an output increment
 - **FD**: final decision for an output increment

Diachronic metrics



- measure change of the incremental hypotheses over time
 - **edit overhead (EO)**: proportion of unnecessary edits
 - **correction time**: (average) difference between FO and FD
 - a confidence for a hypothesis can be derived from its age

Optimising incremental processors

- timing \leftrightarrow diachronic metrics:
 - allow **more right context** \rightarrow better EO, but worse timing
 - **hypothesis smoothing**: only IUs of a certain age are passed on to the next processor



Short recap

- Why build incremental dialogue systems?
- There are some cool incremental dialogue systems already!
- new challenges:
 - self-correction
 - evaluation
 - ...



References

- 1 Incremental dialogue management – basics
- 2 Building an incremental dialogue manager
- 3 Evaluating incremental processors
- 4 References**



References I



Buß, Okko ; Baumann, Timo ; Schlangen, David:

Collaborating on utterances with a spoken dialogue system using an ISU-based approach to incremental dialogue management.

In: Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 2010



Baumann, Timo ; Buß, Okko ; Schlangen, David:

Evaluation and Optimisation of Incremental Processors.

In: Dialogue & Discourse 2 (1) (2011)



Buß, Okko ; Schlangen, David:

Modelling Sub-Utterance Phenomena in Spoken Dialogue Systems.

In: Aspects of Semantics and Pragmatics of Dialogue. SemDial 2010, 14th Workshop on the Semantics and Pragmatics of Dialogue.
2010



Buß, Okko ; Schlangen, David:

DIUM – An Incremental Dialogue Manager That Can Produce Self-Corrections.

In: Proceedings of the 15th Workshop on the Semantics and Pragmatics of Dialogue, 2011



References II



Skantze, Gabriel ; Hjalmarsson, Anna:

Towards incremental speech generation in dialogue systems.

In: Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 2010



Skantze, Gabriel ; Schlangen, David:

Incremental dialogue processing in a micro-domain.

In: In Proceedings of EACL-09, 2009



Thanks for your attention!

