Incremental ASR, NLU and Dialogue Management in the Potsdam INPRO P2 System

Timo Baumann, Michaela Atterer, Okko Buss
IVI Workshop, Bielefeld, 2009-06-08
Context:
Spoken Dialogue Systems

ASR

NLU

dialogue manager

history

domain

TTS

visual output

response generator

[Diagram showing the components of a spoken dialogue system, including ASR, NLU, dialogue manager, history, domain, TTS, and visual output.]
Context: Spoken Dialogue Systems

ASR

NLU

dialogue manager

history

domain

response generator

TTS

visual output

domain
Context:
Spoken Dialogue Systems
Context: Spoken Dialogue Systems

- no reaction before the user finishes talking
Context: **Incremental**

Spoken Dialogue Systems

- **partial results** are being processed immediately
- reaction is quicker, back-channels are possible
Incremental Spoken Dialogue Systems

- **Part 1:** Timo
  - ASR
  - Dialogue Manager
  - NLU
  - TTS
  - Visual output

- **Part 2:** Michaela
  - Partial results are being processed immediately
  - Reaction is quicker, back-channels are possible

- **Part 3:** Okko
  - History
  - Domain

**Context:** Incremental
Context: **Incremental**
Spoken Dialogue Systems

Assessing and Improving Speech Recognition for Incremental SDS: Measures & Methods

Part 1: Timo

Baumann et al., 2009
A Real-World Example of Incremental ASR Hypotheses

- ASR hypotheses change with time (open movie)
A Real-World Example of Incremental ASR Hypotheses

- ASR hypotheses change with time
- more edit than necessary $\rightarrow$ overhead $\sim 90\%$
  - 90% of a consumers work will be useless
A Real-World Example of Incremental ASR Hypotheses

- ASR hypotheses change with time
- more edit than necessary $\rightarrow$ overhead $\sim 90\%$!
A Real-World Example of Incremental ASR Hypotheses

- ASR hypotheses change with time
- more edit than necessary → overhead ~ 90%!
- reduce overhead, sacrifice some timeliness

Patience, Young Jedi! Waiting helps

which edits should we trust?
A Real-World Example of Incremental ASR Hypotheses

- ASR hypotheses change with time
- more edit than necessary \( \rightarrow \) overhead \( \sim \) 90\%
- reduce overhead, sacrifice some timeliness

Software from Malsburg et al., submitted

which edits

waiting helps
A Reduced Example

- $w_{hyp_t}$ is the word sequence hypothesized at time $t$
- two dimensions:
  - time we reason about: $\rightarrow$
  - time we reason at: $\downarrow$
- $w_{gold}$ is final hypothesis
Change Measure

- changes on the right
- *add*, *delete* or *revise*
- ideally: one *add* per word
- in fact: **edit overhead**

$$\text{EO} = \frac{|\text{unnecessary edits}|}{|\text{edits}|}$$
Change Measure

ideally: 3 edits

actually: 11 edits

unwanted: 8 edits

EO: 8/11 = 72 %
Edits are bad:

- edits lead to unnecessary processing of a consumer
  - less edits mean less processing

→ we would like to **reduce the edit overhead**
  → by **deferring** or **suppressing** edits

- deferring edits leads to delays, deteriorating *timing measures* …
Measuring Timing

• when do we find out about a word?
  ▪ word first correct: WFC

• when do we become certain about a word?
  ▪ word first final: WFF

• this is per word
  → averages are important
Measuring Timing

- when do we find out about a word?
  - word first: WFC
  - when do we become certain about a word?
    - word final: WFF
- this is per word
  - averages are important
Timing Measures

- depending on the use-case we may care for …
  - if we want to assume as soon as possible $\rightarrow$ low first
  - if we want to know as soon as possible $\rightarrow$ low final

- deferring edits means two things:
  - worse first timings
    (as the lag passes through)
  - less increase in final timings,
    (if we eliminate wrong edits)
Certainty Considerations

- the correction time for a word is \( \text{WFF} - \text{WFC} \)
- 58.6% of all words are immediately correct
- we can calculate the degree of certainty for given hypothesis ages
- e.g. if a correct hyp. lasts for 0.55s, we can be certain (95%) that it will not change anymore
Improving Incremental ASR

- our primary goal is to reduce edit overhead
- ... by deferring or suppressing edits
  - deferring edits will always hurt first timings
  - less impact on final timings
  - the final (non-incremental) result does not change

→ only trust older parts of hyps. (Right Context)

→ only trust older edits (Message Smoothing)
Right Context to Improve Incremental Performance

- much jitter is at the right end of the hypotheses
  - at time $t$ only evaluate $hyp_t$ up to $t - \Delta$
- we need to take this into account for correctness:
  - $fair$ r-correct: $w_{hyp_{t-\Delta}} = w_{gold_{t-\Delta}}$
- $first$ increases with $\Delta$, $final$ increases $\leq \Delta$
- we can predict the future with negative $\Delta$
  - e.g. fair r-correctness down 50% at 100ms in the future
Message Smoothing to Improve Incremental Performance

• most bad edits only last for a short while
  ▪ "zwei" → "zwar" → "zwei"

→ hold back edits until they reach a certain age
  ▪ only output if they don't die before maturing

• multiple short edits of a word may delay messages:
  ▪ **first** may grow without fixed bounds occasionally
  ▪ probable resolution/mitigation: **future work**
    allow for some kind of "majority smoothing"
Right Context vs. Smoothing

EO parity (50%)

EO

Correctness

Fixed Lag

Smoothing

delay in s (scale shows larger right contexts towards the left)
**Right Context vs. Smoothing**

**Right Context:**
- 530 ms
- Bounded ($\leq 530$)
- Timing increase

**Smoothing:**
- 110 ms window
- Low (+140/67 ms)
- Timing increase

**EO parity (50%)**

Delay in s (scale shows larger right contexts towards the left)
Spoken Dialogue Systems

Part 2: Michaela

Incremental Natural Language Understanding
Incremental NLU in INPRO P2

- RUBISC: Robust Unification-Based Incremental Semantic Chunker (SRSL 2009)
- Incremental Probabilistic Reference Resolver (submitted)
RUBISC: Incremental Chunking with Semantic Chunks

- Chunks based on semantic content rather than syntax
- Inspired by the notion of sense units (Selkirk, 1984)
- φ-phrases: consist of head and all its specifiers/head and all the material on the non-recursive side of the head up to the next head outside of its maximal projection (Nespor and Vogel, 1986)
- Sense units: up to head; semantic chunks: up to semantically relevant material
Domain (revisited)

- Actions:
  - grasp, turn, flip, move
- Objects:
  - w, cross, ...
- End positions
  - head, leg, ...
Incremental Chunking

**Input:**
- turn
- erm
- the
- piece
- erm
- the
- second
- in
- the
- upper
- row
- to
- erm
- clockwise

**Chunk:**
- [turn]
- erm
- erm the
- piece
- piece
- the
- the
- second
- in
- in the
- upper
- row
- to
- to
- to
- erm
- [to erm clockwise]

**Semantics:**
- action: turning
- end: --
- object: [name: --
  xpos: --
  ypos: --]

- action: turning
- end: --
- object: [name: --
  xpos: 2
  ypos: --]

- action: turning
- end: --
- object: [name: --
  xpos: 2
  ypos: 1]

- action: turning
- end: right
- object: [name: --
  xpos: 2
  ypos: 1]
Grammar

@:action
@:entity:name
@:entity:xpos
@:entity:ypos
@:end

action:flipping-> spieg(le|el) \{flip\}
action:grasping,end:empty -> nimm|nehme \{take\}

entity:name:x -> kreuz|plus|((das|ein) x) \{cross|plus|((the|an) x)}
end:horizontal,action:flipping -> horizontal \{horizontally\}
Slot Unification

- time:
  - Input:
    - [schieb]
    - das
    - das mh
    - [das mh horizontal]
    - liegt
    - liegt ins
    - [liegt ins Vorderbein]

- unification component:
  - unify frame with
    - [action:movement]
    - →success:
  - unify frame with
    - [action:flipping]
    - end:horizontal
    - →failed:
  - unify frame with
    - [end:leg1]
    - →success:

- Frame:
  - action:–
    - end:–
    - ...
  - action:movement
    - end:–
    - ...
  - action:movement
    - end:–
    - ...
  - action:movement
    - end:leg1
    - ...
...
Schieb das Kreuz in den Kopf des Tieres, das wie ein Elefant ...

Push the cross into the head of the animal, which looks like....

Atterer, Baumann, Schlangen (Coling 2008): Syntactic features: don't know if sentence has ended

Atterer & Schlangen (SRSL 2009): Chunker state: 3 slots filled:
  action: movement
  object: cross
  end: head
-> can react, barge in etc
Implementation

- OAA-agent
- Supports: add, revoke, commit
- Evaluation: 55% frames correct, 87% slots correct
Incremental Reference Resolver

- Complementary agent to chunker
- Bayesian believe update model which treats the intended referent as a latent variable generating a sequence of observations ($w_{1:n}$ is the sequence of words $w_1, w_2, \ldots, w_n$):

$$P(r|w_1,\ldots,w_n) = \alpha \times P(w_n|r, w_1,\ldots,w_{n-1}) \times P(r|w_1,\ldots,w_{n-1})$$
Reference Resolution -- Disfluencies

• Psycholinguistic evidence: more hesitations when describing something hard (Tanenhaus et al., 1995; Brennan & Schober, 2001; Bailey & Ferreira, 2007; Arnold et al., 2007)

• Include (filled) pauses into our language model
Probabilistic Reference Resolution

- OAA agent demo:
Probabilistic Reference Resolution

- Add 'nimm':

```
simple:
  x
  w

hard:
  f
  n
  y
```

![Bar Chart Image]
Probabilistic Reference Resolution

- Add 'das':

```
  simple:
  x
  w
  hard:
  f
  n
  y
```

![TypeGUI window with 'das' field and bar chart](image)

![Bar Chart](image)
Probabilistic Reference Resolution

- Add '<sil>':

  simple:
  x
  w

  hard:
  f
  n
  y
• Add 'm':

```
simple:
\[ x \]
\[ w \]

hard:
\[ f \]
\[ n \]
\[ y \]
```
Probabilistic Reference Resolution

- Revoke 'm':

```
  simple:
  x
  w

  hard:
  f
  n
  y
```

![TypeGUI interface](image)
## Reference Resolution - Evaluation

- **N-best approach:** select all pieces above a threshold

<table>
<thead>
<tr>
<th></th>
<th>hes</th>
<th>no hes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Last belief correct:</td>
<td>55%</td>
<td>54%</td>
</tr>
<tr>
<td>When correct?</td>
<td>88%</td>
<td>85%</td>
</tr>
<tr>
<td>correct during silences:</td>
<td>37%</td>
<td>31%</td>
</tr>
</tbody>
</table>

(Schlangen, Atterer, Baumann (submitted))
Context: **Incremental**

Spoken Dialogue Systems

Part 3: Okko
Dialogue Manager
Overview

- Receives incremental add/revoke/commit messages from acoustic, ASR and NLU components and allows in- utterance processing.

- Handles timeouts for different kinds of dialogue acts to define behaviour (domain-independent interaction management/dialog skills).

- Delegates actions across modalities via Action Manager
## Dialogue Manager

### Incremental vs transaction-based DM (1)

<table>
<thead>
<tr>
<th>Incremental</th>
<th>Transaction-based</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Events are handled asynchronously.</td>
<td>• Events are handled sequentially.</td>
</tr>
<tr>
<td>• All components are always active, no blocking control.</td>
<td>• Single component has control.</td>
</tr>
<tr>
<td>• Dialogue state and timeouts must vary dynamically.</td>
<td>• Static property set.</td>
</tr>
</tbody>
</table>
Dialogue Manager
Incremental vs transaction-based DM (2)

Transaction-based dialog timeouts

Timers:
- Timeout
- Silence
- Maxspeech
- Silence

User:
- "Take the um..."
- "X-shaped one"

Semantics:
- \{action:-\}
- \{action:'take'\}

System:
- "Which piece?"
- "<takes the X>"
Dialog Manager
Incremental Timers & States (1)

- **Timers**
  - **maxspeech**
    - Cuts off talk that leads nowhere
  - **silence**
    - Has settings for types of silence correlating to dialog states

- **States**
  - **preUtterance**
  - **noContent**
  - **partialContent**
  - **fullContent**
Dialog Manager
Incremental Timers & States (2)

**system:** Which piece? **preUtterance timer**

**user:** <silence>

**system:** Which piece? **preUtterance timer**

**user:** <silence> Erm, yes let's see...

**system:** Which piece? **noContent timer**

**user:** <silence> Erm, yes let's see...

**system:** Which piece? **noContent timer**

**user:** Let's see... take the erm...

**system:** Which piece? **noContent timer**

**user:** Let's see... take the erm...

**system:** Which piece? **partialContent timer**

**user:** Let's see... take the erm...

**system:** Which piece? **fullContent timer**

**user:** Let's see... take the erm...

**system:** Which piece? **fullContent timer**

**user:** Let's see... take the erm...

**system:** Which piece? **fullContent timer**

**user:** Let's see... take the erm...

**system:** Which piece? **fullContent timer**

**user:** Let's see... take the erm...

**system:** Ok! **fullContent timer**

**user:** x-shaped piece
Dialogue Manager
Incremental vs transaction-based DM (3)

Incremental dialog timeouts

user

system

semantics


timers

maxspeech

preUtterance

inUtterance(1)

inUtterance(2)

postUtterance

And now?

Which piece?

Ok.

<takes the X>

And now?

{action:-
  piece:-}

{action:'take'
  piece:-}

{action:'take'
  piece:'x'}

Take the um...

X-shaped one

{action:-
  piece:-}
• Incremental dialog management has the ability to:
  ▪ process user input before end of utterance.
  ▪ reset timers and thresholds during utterance.
  ▪ issue clarification requests mid-utterance.
  ▪ enable backchannel utterances/dialog acts.
Dialogue Manager
Incremental DM State Chart

preUtterance State
- onEntry: Prompt
- onEntry: start silenceTimer[preUtterance]
- onTimeout: Reprompt
- onExit: stop timer

noContent State
- onEntry: start silenceTimer[inUtterance]
- onEntry: start uselessTalkTimer[noSem]
- onTimeout: Reprompt
- onExit: stop timer

fullContent State
- onEntry: start silenceTimer[postUtterance]
- onTimeout: performAction

partialContent State
- onEntry: start silenceTimer[inUtterance]
- onEntry: start uselessTalkTimer[Sem]
- onTimeout: Reprompt
- onExit: stop timer

Synchronization:
- ASR/VAD
- Silence
- Prosody (final)/Chunker (frame)
- Chunker/revoke
Dialog Management

Action Management

- Action Management delegates output to GUI and speech production modalities.
- Aware of status of output modalities (busy/idle), can choose free modality if a dialog act can be embodied in either.
Summing Up
Incremental component features

ASR

Optimization of Incr. Output

NLU

Robust Incr. Chunking, Hesitation sensitive Incr. Statistical RefRes

TTS

visual output

response generator

dialogue manager

status aware, Dynamic modalities

Dynamic thresholds for dialogue states, Backchannel

history

domain
Thank You!

Acknowledgements:

David Schlangen
DFG for funding (Emmy Noether programme)
Transaction-based timeouts (from VXML)

Timers
- Timeout
- Incomplete
- Maxspeech
- Complete

User
- {action:- piece:-}

Semantics
- {action:'take' piece:'x'}

System
- (prompts)
- (responds)