Overview of PRAESINT / CASAM Media Interpretation Agent

- A stream-based media interpretation agent
  - multiple modalities (speech, video, text, audio, video OCR, ...)
  - extended BOEMIE architecture, agent's interpretation loop
    - Assertions / „observations“ arrive via SOAP
    - get accumulated in an Abox (Abox gets bigger and bigger)
      → scalability?
    - Determine what to explain: FIAT generation rules (forward rules)
      → strategy?
    - Explain the FIAT assertion: abduction, extend best interpretation
      → very expensive on big Aboxes, optimization?
    - Collect explanations, probabilistic ranking of interpretations
    - Inform clients about changes in / of the best interpretation
    - Inform clients about alternative interpretations: queries!
ABox Representation of MultiMedia Document

Example: „Text to Audio in local Video“ (Politician to Speech)

MultiMediaDocument

AudioContent

AudioSegment

Locator

Speech

Start 0'05"
End 0'15"

VideoContent

VideoSegment

overlaps

belongsTo

TextContent

TextSegment

Politician

CharStart 10
CharEnd 22

CharStart 0'03"
CharEnd 0'25"
ABox Representation of MultiMedia Document

Example: „Text to Audio in local Video“ (Politician to Speech)

- AudioContent
- VideoContent
- TextContent

AudioSegment overlaps VideoSegment

CharStart to CharEnd

Speech

0'05'' to 0'15''

0'03'' to 0'25''

10 to 22

Locator

MultiMediaDocument hasInterpretation

Politician

belongsTo

PoliticalInterview

Start, End

CharStart, CharEnd
Identification of Coocurrences: Text X Audio

(defquery text-to-audio-in-local-video (?x ?y)
  (and
    (?mmd #!mco:MultimediaDocument)
    (?mmd ?vc #!mco:hasLogicalDecomposition)
    (?vc #!mco:VideoContent)
    (?vc ?vs #!mco:hasMediaDecomposition)
    (?vs #!mco:VideoSegment)
    (?vs ?vl #!mco:hasSegmentLocator)
    (not (?vs #!mco:GlobalVideoSegment))
    (?mmd ?tc #!mco:hasLogicalDecomposition)
    (?tc #!mco:TextContent)
    (?tc ?vs #!mco:belongsTo)
    (?tc ?ts #!mco:hasMediaDecomposition)
    (?mmd ?ac #!mco:hasLogicalDecomposition)
    (?ac #!mco:AudioContent)
    (?ac ?as #!mco:hasMediaDecomposition)
    (?as #!mco:AudioSegment)
    (?as ?al #!mco:hasSegmentLocator)
    (?al ?sm #!mco:overlaps)
    (?sm #!mco:SegmentLocator)
    (?sm ?vl #!mco:overlaps)
    (lambda (audio-near-video-p ?al ?vl "00:00:05,000"))
    (?ts ?x #!mco:depicts)
    (?as ?y #!mco:depicts)
    (not (?x ?y same-as))))
Fiat Generation

Coocurrence of „Politician“ and „Speech“ → FIAT rule fires

MultiMediaDocument

AudioContent

VideoContent

TextContent

AudioSegment

VideoSegment

TextSegment

Locator

Speech

politicianToSpeech

Politician

Start      End

CharStart  CharEnd

0'05'' 0'15''

0'03'' 0'25''

10      22
(define-rule (?x ?y #!edo:politicianToSpeech)
  (and
   (?x #!edo:Politician)
   (?y #!edo:Speech)
   (?x ?y text-to-audio-in-local-video)
   (?x nil #!edo:politicianToSpeech)
   (nil ?y #!edo:politicianToSpeech))
:backward-rule-p nil)
Explanation

Explanation of FIAT politicianToSpeech: PoliticalInterview

MultiMediaDocument

AudioContent

VideoContent

TextContent

AudioSegment

VideoSegment

TextSegment

Overlap

Speech

politicianToSpeech

Politician

Locator

Start: 0'05''

End: 0'15''

Speech

Start: 0'03''

End: 0'25''

Politician

CharStart: 10

CharEnd: 22
(define-rule (?x ?y #!edo:politicianToSpeech)
  (and
   (?x #!edo:Politician)
   (?y #!edo:Speech)
   (?z #!edo:PoliticalInterview)
   (?z ?x #!mco:builtFrom)
   (?z ?y #!mco:builtFrom))
  :forward-rule-p nil)
Link Interpretation Result to VideoSegment

Explanation of FIAT politicianToSpeech: PoliticalInterview

AudioContent

VideoContent

TextContent

AudioSegment

VideoSegment

TextWriter

Politician

Speech

Locator

CharStart  CharEnd

Start      End

Start      End

Start      End

0'05''  0'15''

0'03''  0'25''

10  22

belongsTo

hasInterpretation

politicianToSpeech

 overlaps
Problems & Solutions – 1

- **Scalability**
  - Cooccurrence identification with defined queries, e.g. text-to-audio-in-local-video
    - results are not cached
    - Fiat rules get very complex (def. queries treated by unfolding!)
    - relationships were found / computed again and again
    - cooccurrences have to be found again for linking the interpretation result to the VideoSegment

_Solution: prepare the Abox before processing_

- Establish links from VideoSegments to all „cooccurring“ EDO concept instances
- cooccurrence only computed once and result stored
- EDO information per Segment directly available and explicit
Link Interpretation Result to VideoSegment

Explanation of FIAT politicianToSpeech: PoliticalInterview

MultiMediaDocument

AudioContent

AudioSegment

0'05'' 0'15''
Start   End

VideoContent

VideoSegment

0'03'' 0'25''
Start   End

TextContent

TextSegment

10 22
CharStart  CharEnd

Overlap:

Locator

Speech

Politician

belongsTo

ourDepicts
Optimized Fiat Rule „politicianToSpeech“

(define-rule (?x ?y #!edo:speechToPolitician)
  (and
   (?vs ?x #!mco:ourDepicts)
   (?vs ?y #!mco:ourDepicts)
   (?x #!edo:Speech)
   (?y #!edo:Politician))
  :backward-rule-p nil)
Problems & Solutions – 2

- Realization of Incrementality („stream-based“)
  - for new assertions, identify the affected part of the Abox!
    - add new assertions to global Abox
    - check if there are changes in the „ourDepicts“ relations
    - for affected segments, collect required assertions for interpret.
      → 80% reduction

- Multiple interpretations (many!)
  - how to keep the interpretation Aboxes small?
    → decouple the big common part of the interpretations

- Optimization of abduction

- Query generation problem
  - How to inform the client about alternative interpretations?
RMI Implementation of `receiveAssertions`

- Manage agenda (updates, query answers, ...)
- Abox augmentation
- Determine focus, compute relevant part of CP ABox

Perform the abduction in a loop until termination criterion met (max. # fiats, no more fiats, no probability increase, ...)

KDMA
Assertion Set
Input Processor
Interpretation Processor
RMI Interpretation Engine

Query
Interpretation
RMI Input Processor

Assertion Set

Hypothesized Assertions + hasInterpr.

RMI Agenda

Common Part (CP)

Identify Assertions

Add / rem. / mod. CP

Remove incon. Int.

TBox

Identify Query Answ.

Del. incomp. Interpret.

Sort Agenda

Commun. Changes

Identify relevant CP Subset

Identify affected Segments

Add Segments for EDOs

Reject Ass. with low Cert.

„Database“ queries /rules have to operate on FULL CP!

20%
RMI Interpretation Processor

Terminate? No -> Fiats for = {} Yes

Fiats for = {} No Apply Fiat rules to Reduce gen. Fiats

Yes

Commun. Changes

Best

Fiat Rules

Strategy

Select Fiat for

Abd. Rules

Best

Best

Abduction Explain Fiat on

Best

Prob. Rules

Sort Agenda

Best

Sort

Agenda

rrel. CP

1 ... k

1 ... k ... n

...one per type & video seg.
RMI Communicate Changes

Augment best w. blank relational Structure

Added: Best \ Prev.Best

Removed: Prev.Best \ Best

Best  Previous Best

Send Interpretation
ActionT.: remove

Send Interpretation
ActionT.: add

Yes

Added = {}

No

Yes

Remove = {}

Prev.Best ← Best

Create Queries

Yes

No

required for Queries!
identify
Key Assertions
for first m (≥ k)
Interpretations

\[ \exists_1 \cdots \exists_k \cdots \exists_m \]

Create
OR
Query

End
Computation of Queries

- Computation of characteristic ("key") assertions $\Xi_i$ for $\Delta_i, 1 \leq i \leq n$

- Compute the "common differences" by intersecting all differences to all other $\Delta_j$

$$\Xi_i = \bigcap_{i \neq j, 1 \leq j \leq n} \Delta_i \setminus \Delta_j$$

- From each $\Xi_i$ select an assertion (preferable an instance assertion)
  
  $\rightarrow$ n disjuncts for OR query  
  $\rightarrow$ simple score: $1 - 1/n$

- "\" may be ABox difference, but...
What is the blank relational structure and why is it required?

- Problem:
  - queries can only be formulated against the communicated „best“ interpretation: $\Delta_i$
  - However, all but one query disjuncts come from $\Xi_j \subseteq \Delta_j$
  - the relational structures may be completely different
    - different hypothesized RMI INDs, different edges, etc.

- Example: how to communicate the difference between

  - HCI only knows Ind1!
  - Q-Disjunct1: Ind1 : Person
  - Q-Disjunct2: Ind1 : Interview
    Ind1 : Interviewer

- Solution: avoid the problem in the first place!
What is the blank relational structure and why is it required? (2)

• Instead of only sending the best interpretation, we also include the „blank relational structure“ of ALL other interpretations

→ relational structure and all hypothesized INDs known to HCI

- HCI knows Ind1, Ind2, Ind3!
- Q-Disjunct1: Ind1 : Person
- Q-Disjunct2: Ind2 : Interview
  [ Ind3 : Interviewer ]
  [ (Ind2, Ind3) : b.F. ]
- No „new-ind mapping“ needed
Abductive Query Answering

- Simple example
  - Query: \( \text{ans}(x) \leftarrow C(x), D(y), R(x, y) \)
  - Abox: \( \{(i, j) : R, i : C\} \)
  - **Preferred** solution (optimal, according to score defined below)
    \[
    x \leftarrow i, y \rightarrow j : \\
    \Delta = \{j : D\}
    \]
  - **Other** solution (plus 7 more, \(3^2 = 9\)), e.g.
    \[
    x \leftarrow \text{new}_1, y \leftarrow \text{new}_2 : \\
    \Delta = \{\text{new}_1 : C, \text{new}_2 : D, (\text{new}_1, \text{new}_2) : R\}
    \]
- Exponential number of solutions has to be computed to find „the best“
  - **optimization idea**: early dynamic cutoff of search space based on score evaluation on partially computed explanations (deltas)
„Depth First“ Abductive Query Evaluation

\[ \mathcal{A} = \{(i, j) : R, i : C\} \]

Partial Delta

\[ \text{new}_1 \]

\[ \text{new}_2 \]

Leaf = compl. Delta

Query Evaluation Plan
CASAM Preference Score

Very simple:

entailed Assertions minus hypothesized Assertions

\[
\text{score}(\Delta) =_{def} |\Delta^+| - |\Delta^-| \rightarrow \text{maximize}
\]

\[
\Delta = \Delta^+ \cup \Delta^-(\text{entailed, hypothesized})
\]
Illustrations of (Partial) Scores

\[ A = \{(i, j) : R, i : C\} \quad |\Delta^+| - |\Delta^-| = \text{score} \to \text{max.} \]

```
<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>\text{-1} &amp; new_1 &amp;   &amp;   &amp;   &amp;   &amp; C(x)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\text{-2} &amp; y &amp; 0 &amp; y &amp; y &amp; y &amp; D(y)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>new_2 &amp; i &amp; j &amp; new_2 &amp; i &amp; j &amp; R(x, y)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>\text{-3} &amp;   &amp;   &amp;   &amp; 1 &amp;   &amp;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ \Delta_1 \quad \Delta_4 \quad \Delta_6 \quad \Delta_9 \]

\[
\begin{align*}
0 - 3 &= -3 \\
1 - 2 &= -1 \\
2 - 1 &= 1 \\
0 - 3 &= -3
\end{align*}
\]
Score-Based Cutoff of Search Space

\[ A = \{ (i, j) : R, i : C \} \]

Rem. points I can make: 2
-1 + 2 = 1 → continue (may be as good as B.S.F)

Rem. points I can make: 1
-2 + 1 = -1 → CUTOFF (is worse than B.S.F)

Best so far
\[ \Delta_6 \]
2 - 1 = 1

\[ \Delta_1 \]
0 - 3 = -3

CAN PRUNE WHOLE SUBTREES!
More formally...

\[ n = |\Delta^+| + |\Delta^-| \quad (n \text{ const. for each rule body}) \]

\[ \text{score}(\Delta) =_{def} |\Delta^+| - |\Delta^-| \rightarrow \text{maximize (not monotone)} \]

\[ n + \text{score}(\Delta) = 2|\Delta^+| \]

\[ \text{score}(\Delta) = 2|\Delta^+| - n \rightarrow \text{maximize (and monotone!)} \]

- Let \( \Delta_p \subseteq \Delta, m_p = n - |\Delta_p| \) (remaining conjuncts)
  - If \( \text{score}(\Delta_p) + (n - |\Delta_p|) < \text{score}(\Delta_{best\_so\_far}) \)
    \[ \text{score}(\Delta_{best\_so\_far}) - \text{score}(\Delta_p) > (n - |\Delta_p|) \]
    reject \( \Delta_p \)
How Effective is this?

- Synthetic benchmark: finding graph isomorphisms (n nodes)
- Problem reductions:
  Graph Isomorphism → ABox Difference → Abduction

![Diagram showing isomorphism and problem reduction](image)

<table>
<thead>
<tr>
<th># Nodes in Ring</th>
<th>Seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>10</td>
<td>8</td>
</tr>
<tr>
<td>12</td>
<td>10</td>
</tr>
<tr>
<td>14</td>
<td>12</td>
</tr>
</tbody>
</table>

**Optimized**

**Unoptimized**
Appreciation of Complexity

• Some numbers
  
  – video 6, after bunch 3: 283 Fiats (new rule set)
  
  • potential quadratic number of Fiats (in terms of inds in the Abox)
  
  • after reduction „only one Fiat per type and shot“: 46 Fiats
  
  – „external complexity“ of interpretation loop
    
    • each Fiat may generate 2 to 3 explanations
    
    • branching will easily kill the system
  
  – „internal complexity“ of abduction (hidden in RacerPro)
    
    • in order to find these 2 to 3 best explanations PER FIAT, yet another exponential number of explanations has to be considered!
    
    • exponential in the number of individuals in the ABox

→ RMI handles serious complex problems, more must be done for meta reasoning (we stop after 30 Fiats per bunch)
Open Issues

- Reimplementation of probabilistic valuation and
- React to removed / confirmed tags
- React to „negative“ query answers
  - only positive query answers considered so far
  - „shuffle“ the interpretations containing the answer assertions to the front of the agenda
- More specific Fiat generation rules
- Anytime / meta reasoning
  - reduce set of assertions if timeout occurs, etc.
  - some dumb strategies already implemented
- Q: do we really have to keep all interpretations on the agenda?