Computer-Aided Semantic Annotation of Multimedia

Plenary Meeting: WP-3
Oliver Gries, Ralf Möller,
Maurice Rosenfeld, Michael Wessel
TUHH
Tasks

- Task 3.1: Optimized reasoning engine for probabilistic first-order structures (Lead TUHH)
  - New approach developed (Paper presented by Oliver at UniDL’10)

- Task 3.2: System supporting probabilistic abduction as a reasoning service (Lead TUHH)
  - Anahita presents paper at RR 2010
  - Michael’s presentation

- Task 3.4: Meta-level reasoning component (Lead TUHH)
  - Query generation integrated into second prototype (Michael’s pres)
  - See upcoming deliverable D3.4

- Task 6.2: MM Ontology: MESH ontology
New RMI Implementation

• Overview (I can skip slides on request)
  – Implemented architecture
  – Computation of queries
  – Optimization of abduction
  – Open issues

• CASAM Team @ STS / TUHH
  • Anahita Nafissi
  • Oliver Gries
  • Ralf Möller
  • Maurice Rosenfeld
  • Kamil Sokolski
  • Michael Wessel
What's new in a Nutshell...

- Agenda-based
  - manages RMI interpretations as small individual ABoxes
    + big „common part“ ABox CP (segments, EDO/MCO stuff, …)
  - incremental: **only reinterprets what needs to be reinterpreted**
    - uses only the relevant subset of CP (20% of CP) for Fiat rules
    - abduction performed on **subset** of CP + best interpretation
      → even „higher levels“ of interpretation possible
  - more control on interpretation process, by looking at the agenda
    (more information explicitly available) → **meta level reasoning**

- Queries computed for interpretations on agenda

- Lisp-based & multi-core ready
  - shares memory structures with RacerPro (no more OWL-in-out)
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 → Interpretation

HCI

RMI Interpretation Engine

Query
RMI Input Processor

- Hypothesized Assertions + hasInterpr.
- Common Part (CP)
- RMI Agenda
- Identify Assertions
- Add / rem. / mod. CP
- Remove incon. Int.
- TBox
- Identify Query Answ.
- Del. incompl. Interpret.
- Sort Agenda
- Commun. Changes
- Reject Ass. with low Cert.
- Identify relevant CP Subset
- Identify affected Segments
- Add Segments for EDOs
- "Database" queries /rules have to operate on FULL CP!
- our-Depicts Rules
- ABox Queries

20%
Apply Fiat rules to RMI Interpretation Processor

Fiat for \( \{ \} \)

Select Fiat for

Abduction Rules

Commun. Changes

Fiat Rules

Reduce gen. Fiats

Sort Agenda

Strategy

Best

Abd. Explain Fiat on

Best

Prob. Rules

Fiat on

Best...

one per type & video seg.

Yes

No

Yes

No

Terminate?

Fiats Agenda

1 \ldots k

Rel. CP

1 \ldots k \ldots n

Best

Best

Best

Best

Best

Best

Best

Best

Best

Best

Best

Best

Best

Best

Best
RMI Communicate Changes

Augment best w. blank relational Structure

Added: Best \ Prev. Best

Removed: Prev. Best \ Best

Send Interpretation
ActionT.: remove

Yes

Removed = {} No

Send Interpretation
ActionT.: add

Yes

Added = {} No

Create Queries

Prev. Best ← Best
RMI Create Queries

- Identify Key Assertions for first \( m \geq k \)
- Interpretations

\[ \Xi_1 \cdots \Xi_k \cdots \Xi_m \]

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 : 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: \( ans() \leftarrow C(x), D(y), R(x, y) \)
  - 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

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

\[ \text{Partial Delta} \]

\[ \text{Leaf} = \text{compl. Delta} \]

\[ \text{Query Evaluation Plan} \]

1. \( G \)
2. \( G \)
3. \( T \)
CASAM Preference Score

Very simple:
entailed Assertions minus hypothesized Assertions

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

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

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

-1 new1

C(x)

-2 y

D(y)

0 y

R(x, y)

| \Delta_1 = 0 - 3 = -3 | \Delta_4 = 1 - 2 = -1 | \Delta_6 = 2 - 1 = 1 | \Delta_9 = 0 - 3 = -3 |
Score-Based Cutoff of Search Space

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

\[ new_1 \]

\[ i \]

\[ j \]

\[ x \]

\[ -1 \]

Rem. points I can make: 2

\[-1 + 2 = 1 \rightarrow \text{continue} \]

(may be as good as B.S.F)

\[ Rem. \text{ points I can make: } 1 \]

\[-2 + 1 = -1 \rightarrow \text{CUTOFF} \]

(is worse than B.S.F)

CAN PRUNE WHOLE SUBTREES!

\[ \Delta_1 \]

\[ 0 - 3 = -3 \]

Best so far

\[ \Delta_6 \]

\[ 2 - 1 = 1 \]

Rem. points I can make: 1

\[-2 + 1 = -1 \rightarrow \text{CUTOFF} \]

(is worse than B.S.F)

CAN PRUNE WHOLE SUBTREES!
More formally...

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

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

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

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

- 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

---

**Graph**: ABox A

**Graph**: ABox B

**Diagram**:
- Optimized vs Unoptimized performance
- # Nodes in Ring vs Seconds

**Chart**:
- Y-axis: Seconds
- X-axis: # Nodes in Ring
- Optimized: Blue squares
- Unoptimized: Red diamond

**Legend**:
- Optimized
- Unoptimized

---

**Cloud Note**:
- isomorph
- <= diff. empty
- <= max. score
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 individials 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?