How to enhance genericity in natural language command interpretation using introspection and ontologies?

Laurent Mazuel

Advisor: Nicolas Sabouret

10/05/07 – SpeD'2007
Plan

- Motivation
- The Solver Language: Introspection
- Natural Language Processing & Solver language
- Description & example
- Evaluation
- Conclusion & Future works
Framework: NL command system

Flight Reservation by Phone
- Domain A
- Solver A
  - NL Syntax A
  - NL Interpretation A
  - NL Generation A

Medical Semantic Search Engine
- Domain B
- Solver B
  - NL Syntax B
  - NL Interpretation B
  - NL Generation B

Generic NL Algo
Genericity: Related Works

[Dzikovska04]: Generic parser, *but* ad-hoc ontology formalism
[Milward04]: Generic ontology, *but* specific NLP modules
[Eliasson07]: Generic solver language, generic NLP algorithms, *but* very specific ontology formalism (syntax+semantic)

Claim: need for agent's introspection

NLP Only depends on agent's code + ontology
Solver: The VDL language

- Term rewriting => introspection
- Formal Commands => XML nodes

Term rewriting at each execution step
Formal Actions description

- **Preconditions:**
  - Event ($P_e$) Ex: “drop” or “take”? 
  - Structure ($P_s$) Ex: event has “object” param? 
  - Context ($P_c$) Ex: robot has enough energy? 
  - Context/Structure ($P_{cs}$) Ex: the “object” exists in world? 

- **Effects:** nodes addition and deletion
Natural Language Processing & VDL

User → NLP → Embodied Conversational Agent

introspection → OWL Ontology
Example: Jojo

User:
“drop the circle on the lower line”
NLP Architecture

Tagger
Chunker
Lemmatizer

Dialogue Manager

XML-based NL Generator

Concept Matching

OWL Agent

Events Generation introspection

Events Selection
Concepts matching

User Concepts

VDL Concepts

Agent Ontology

Matching concepts

Ontology: sameAs & Identity
NLP Architecture

- Tagger
- Chunker
- Lemmatizer
- Dialogue Manager
- XML-based NL Generator
- Concept Matching
- Events Selection
- OWL Agent
- Events Generation
- introspection
Events generation

• Main ideas
  – Testbed generation based approach
  – Precondition analysis

• Output
  – The set E of possible events
    \[ E = \{ \text{refine}(e, r), \forall e \in P_{e+c}, \forall r \in R_e(e) \mid \forall p \in P_s(r) \cup P_{cs}(r), \text{eval}(p) = \top \} \]
  – The set F of currently impossible events
    \[ F = \{ \text{refine}(e, r), \forall e \in P_e, \forall r \in R_e(e) \mid \forall p \in P_s(r), \text{eval}(p) = \top \} \]
Example

Events of E:
• 8 events:
  – 7 for the grid status
  – 1 for “outer-box”

Example:
<drop><position>
  <outer-box/>
</position></drop>

Events of F:
• 9 events:
  – 7 for « take »
    (the « hand » is full)
  – 2 for « drop »
    (the cell is full)
NLP Architecture

- Tagger
- Chunker
- Lemmatizer

- Dialogue Manager

- Concept Matching

- XML-based NL Generator

- Events Selection

- OWL Agent
  - introspection

- Events Generation

- NL Generator
Events selection

- For all events of E & F: *relevance score*
  \[ p(e) \& p(f) \in [0, 1] \text{ dependant of user concepts } C \]
  
  \[ p(e) = \frac{\text{card}(\{c \in C | \text{contains}(e, c)\})}{\text{card}(C)} \]

- **Output**: The sub-set \( \mathcal{E} \& \mathcal{F} \) with *maximal relevance score*

\[
\mathcal{E} = \begin{cases} 
\emptyset & \text{if } \max(\{p(e), e \in E\}) = 0 \\
\{e \in E | p(e) = \max(\{p(e), e \in E\})\} & \text{otherwise}
\end{cases}
\]
Example

- C={\textit{drop, circle, lower}}
- Relevance of events in E:
  - e="drop lower center/right": \(p(e)=2/3\)
  - e="drop middle/upper XXX": \(p(e)=1/3\)
- Relevance of events in F:
  - f="take XXX circle": \(p(f)=1/3\)
  - f="take XXX": \(p(f)=0\)
  - f="drop lower left": \(p(f)=2/3\)
  - f="drop middle": \(p(f)=1/3\)
NLP Architecture

Dialog Manager

Tagger
Chunker
Lemmatizer

XML-based NL Generator

Concept Matching

Events Selection

Events Generation

OWL Agent

introspection
Dialogue manager

- Answer to the user depending on $\mathcal{E}$, $\mathcal{F}$, $p(e)$ & $p(f)$.

- Two threshold $p_{\text{min}} = 0.3$ & $p_{\text{max}} = 0.8$

  1) $p(e)$ & $p(f)$ regards to $p_{\text{min}}$ et $p_{\text{max}}$

  2) $|\mathcal{E}|$ & $|\mathcal{F}|$

- Example: $p(e) = 0.9$ & $p(f) = 0.5$, events in $\mathcal{E}$ are relevant

- Example: $p(e) = 0.5$ & $p(f) = 0$, events in $\mathcal{E}$ are not sure

- Example: $p(e) = 0.7$ & $p(f) = 1$, impossible user command, but $\exists$ relevant events possible
Example

• \(|\mathcal{E}| = 2 \& p(e) = 2/3\)
• \(|\mathcal{F}| = 1 \& p(f) = 2/3\)

Dialogue manager case:

\[ p_{\min} \leq p(e) \leq p_{\max} \quad \& \quad p_{\min} \leq p(f) \leq p_{\max} \]

\[ p(f) \leq p(e) \]

Command is understood but no sure and with ambiguity
Example: final result

Welcome to 1-jojo's page

Hand:

Hand:

Hand:
Evaluation protocol

- 8 subjects
- No guideline:
  - First use
  - No NL information
  - No time limitation
- Questionnaire
Evaluation main results

● Main remarks
  +++: Feedback on Possible Events
  ++: Explanations on error
  +-: Semantic difficulties

● Finally,

  \textit{Introspection} +
  \textit{Semantic} -
Conclusion

- Algorithm only parametrized by:
  - Agent code
  - Domain ontology
- Easily reusable
- First evaluation
Future works

User:
“put the green cube where the grey circle is”

Pb 1: Semantic Similarity
Pb 2: “Bag of words”
Pb 3: Lexical
Pb 4: Goal, planning
Questions?