Automated planning for situated natural language generation

Konstantina Garoufi and Alexander Koller
Cluster of Excellence on Multimodal Computing and Interaction
Saarland University
Motivation and goal

Our goal

1. walk forward and go through the door
2. (... ok and then turn to your right
3. and then push the left red button

push the left red button on the right wall in the next room in front of you
Outline

• Motivation and goal

• Context manipulation for situated natural language generation

• Generating context-dependent adjectives

• Evaluation and conclusions
The problem

How to navigate the user so that the forthcoming referring expression is cognitively simple?

Push the button on the wall to your left.

Push the button on the wall to your left in the next room.
Earlier approaches

Stoia et al. (2006)

• monitor non-linguistic context changes to estimate when conditions become appropriate for generating references

• but cannot predict how to deliberately change the context in appropriate ways
Earlier approaches

Planning-based generation

• provides the foundations for this work, which builds upon the sentence generation as planning approach (Koller & Stone, 2007)
Sentence generation as planning

The white rabbit sleeps.

Koller & Stone (2007)
Our approach

actions have non-linguistic effects

only visible objects can be manipulated

will explain later

sentence conjunction connects two events

**action: turn-left(e,x,y)**

**precondition:** player-ori(x), ...

**effect:** ¬player-ori(x), player-ori(y), ...

**action: push(e,x,p,o)**

**precondition:** visible(p,o,x), ...

**effect:** ∀y. (y≠x ∧ visible(p,o,y)
→ distractor(y), to-do(push(x)), ...

**action: and(e1,e2)**

**precondition:** ...

**effect:** ...

---

S:self

V:self

Adv

turn

left

---

S:self

V:self

NP:obj ↓
push

---

S:self *

and

S:other ↓
Solution

Turn left

turn-left(e1,o1,o2)

Action: turn-left(e1,o1,o2)

Precondition: player-ori(o1), ...

Effect: ¬player-ori(o1), player-ori(o2), ...

State
player-pos(p1)
player-ori(o1)
¬visible(p1,o1,b1)
Solution

State
player-pos(p1)
player-ori(o2)
visible(p1,o2,b1)

action: and(e1,e2)
precondition: ...
effect: ...

action: the-button(b1)
effect: ∀y. (¬button(y) → ¬distractor(y)), ...

action: push(e2,b1,p1,o2)
precondition: visible(p1,o2,b1), ...
effect: ∀y. (y ≠ b1 ∧ visible(p1,o2,y) → distractor(y)), to-do(push(b1)), ...

and push the button.

and(e1,e2)
push(e2,b1,p1,o2)
the-button(b1)
Solution

Push the button.

State
todo(push(b1))
Outline

- Motivation and goal
- Context manipulation for situated natural language generation
  - Generating context-dependent adjectives
- Evaluation and conclusions
The problem

How to predict the user’s interpretation of context-dependent expressions like “left”?
Earlier approaches

van Deemter (2006)

• generates references that can contain one gradable modifier

• but cannot handle multiple context-dependent modifiers
Our approach

<table>
<thead>
<tr>
<th>action: left(x)</th>
</tr>
</thead>
<tbody>
<tr>
<td>precondition: ∀y. ¬(distractor(y) ∧ left-of(y,x)), ...</td>
</tr>
<tr>
<td>effect: ∀y. (left-of(x,y) → ¬distractor(y)), ...</td>
</tr>
</tbody>
</table>

captures presupposition that no current distractor is to the left of the target

rules out from the set of distractors anything that is to the right of the target

target referent

distractors
Solution

Push the left red button.

push(e,b2,p1,o3)
the-button(b2)
red(b2)
left(b2)

action: the-button(b2)
effect: ∀y. (¬button(y) → ¬distractor(y)), ...

action: push(e,b2,p1,o3)
precondition: visible(p1,o3,b2), ...
effect: ∀y. (y ≠ b2 ∧ visible(p1,o3,y) → distractor(y)), to-do(push(b2)), ...

State
player-pos(p1)
player-ori(o3)
visible(p1,o3,b2)

action: left(b2)
precondition: ∀y. ¬(distractor(y) ∧ left-of(y,b2)), ...
effect: ∀y. (left-of(b2,y) → ¬distractor(y)), ...

action: red(b2)
effect: ∀y. (¬red(y) → ¬distractor(y)), ...

Push the left red button.

target
Solution

Push the left red button.
Other contributions

• We enforce the class-based premodifier ordering paradigm (e.g. Mitchell, 2009) for linguistically acceptable referring expressions

  *the red left button* ✗
  *the left red button* ✔

• We argue for a novel scalable model of planning perlocutionary acts (Koller et al., SemDial 2010)
Outline

• Motivation and goal
• Context manipulation for situated natural language generation
• Generating context-dependent adjectives
• Evaluation and conclusions
Evaluation framework: GIVE Challenge

User plays 3D game in virtual world.

Natural language generation system generates instructions in real time.

Koller et al. (2009)
Our system SCRISP

- We solve our planning problems with off-the-shelf FF planner (Hoffmann & Nebel 2001; Koller & Hoffmann, 2010)
- Max planning time 1.03 sec (3 GHz CPU) for a knowledge base of about 1500 facts and a grammar of about 30 lexicon entries

1. Turn right and move one step.
2. Push the right red button.
Baseline A

1. Push the right red button on the wall to your right.

- Fully describes target with respect to user’s current location
- Does not attempt to manipulate context
Baseline B

- Improved version of GIVE-I Challenge’s best-performing system (Chen & Karpov, 2009)
- Treats all adjectives as intersective, based on Areces et al. (2008)
- Starts off with navigation and opportunistically describes target once it can

1. Turn right.
2. Walk forward 3 steps.
3. Turn right.
4. Walk forward 1 step.
5. Turn left.
6. Good! Now press the left button.
Results

SCRISP is more successful than Baseline A and comparable to Baseline B.

SCRISP’s references are more successful than Baseline A’s but less than Baseline B’s.

<table>
<thead>
<tr>
<th></th>
<th>overall success</th>
<th>reference success</th>
<th>reference distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCRISP</td>
<td>69%</td>
<td>71%</td>
<td>2.49</td>
</tr>
<tr>
<td>Baseline A</td>
<td>16%**</td>
<td>49%**</td>
<td>1.97*</td>
</tr>
<tr>
<td>Baseline B</td>
<td>84%</td>
<td>81%*</td>
<td>2.00*</td>
</tr>
</tbody>
</table>

Differences to SCRISP are significant at *p < .05, **p < .005.

however SCRISP can generate references from further away.
Conclusions

• Automated planning for situated generation can flexibly model different phenomena:
  • generation of instructions that deliberately manipulate the context in convenient ways
  • generation of references with context-dependent modifiers
• With modern automated planning technology this is possible in real time
Thank you!