Natural Language Generation for Embodied Conversational Agents

Day 3

Kristina Striegnitz

ESSLLI 2008
Hamburg, Germany

Yesterday

Yesterday - BML specifications

<speech id="s">
  and now take <sync id="t1"/> this bar and make it <sync id="t2"/> this big <sync id="t3"/>
</speech>

<gesture id="g1" type="POINT" target="obj" stroke="s:t1"/>
<gesture id="g2" type="GENERIC" stroke-start="t2" stroke-end="t3"
  hand="both"
  two-handed="mirror"
  handshape=open hand"
  location="center, center, medium"
  orientation="palm inward, finger forward"
/>
Today - Where do those representations come from?

From communicative goal to BML representation

- **canned text based**
  - goal → canned text → text + non-verbal behavior (BML)
  - domain knowledge
dialogue context

- **grammar based**
  - goal → ? → semantic content of sentence → text + non-verbal behavior (BML)
  - grammar
domain knowledge
dialogue context

Excursion: LTAG - Lexicalized Tree Adjoining Grammar

- Mapping semantics to syntax
- Content determination
- Referring Expression Generation
  - multimodal referring expressions
    - generating pointing gestures
    - generating iconic gestures

Cassell, Vilhjalmsson & Bickmore 2001
Lee & Marsella 2006
**LTAG with semantics and pragmatics**

Stone et al. 2003  
Koller & Stone 2007

![Diagram of LTAG with semantics and pragmatics](image)

**Mapping semantics to syntax**

To communicate: like(e,m,r) name(m,mary) rabbit(r) white(r)  
Discourse context: hearer-old(r)  
Domain knowledge: animate(r)

![Diagram of mapping semantics to syntax](image)

**Integrating gestures**

[Cassell, Stone & Yan 2000]

Structure for synchronizing gestures with syntactic phrases:

- **SYNC**
  - G
  - C

  - Gesture synchronized with gesture

Example lexical entry requiring a gesture:

- **Syntax:**
  - S
    - NP
    - VP
      - V
      - SYNC

  - Semantics: have(o,x)
  - Pragmatics: hearer-new(x) ∧ theme(o)

**SPUD - lexical entries for gestures**

A "word" entry with the same semantics. Gestures can be semantically redundant or complementary:

- **Syntax:**
  - NP
  - VP
    - V

  - Semantics: surround(x,p)

- **Syntax:**
  - NP
  - VP

  - Semantics: surround(x,p)
SPUD - building a multi-modal utterance specification

Where does the semantics come from?

- grammar based

For example:
  
  goal: describe how to get from point A to point B

  ... [turn(right, b1), building(b1), tall(b1)] ...

Content determination example: walking directions

User: how do I get from building A to building B?
Communitive goal: describe how to get from point \( p_a \) to point \( p_b \).

Today

- Mapping semantics to syntax
- Content determination
- Referring Expression Generation
- multimodal referring expressions
  - generating pointing gestures
  - generating iconic gestures

Kristina Striegnitz, Union College - ESSLI 2008 Aug 11-15
User: how do I get from building A to building B?
Communicative goal: describe how to get from point $p_a$ to point $p_b$.

- $p_a, p_{49}, p_{50}, p_{58}, p_{63}, p_{80}, p_{81}, p_b$
- start $l_m_a$
- end $l_m_a$

- A* search
- determine reorientation points
- pick landmarks for reorientation points
- pick landmarks for long straight segments

---

User: how do I get from building A to building B?
Communicative goal: describe how to get from point $p_a$ to point $p_b$.

- $p_a, p_{49}, p_{50}, p_{58}, p_{63}, p_{80}, p_{81}, p_b$
- start $l_m_a$
- right $l_m_c$
- left $l_m_b$
- end $l_m_b$

- A* search
- determine reorientation points
- pick landmarks for reorientation points
- map to a sequence of messages
Content determination example: walking directions

User: how do I get from building A to building B?

Communicative goal: describe how to get from point \( p_s \) to point \( p_a \).

\[
\begin{align*}
p_a & \text{ p49, p50, p58, p63, p80, p81, p_a} \\
\text{start} & \rightarrow \text{right} \\
\text{lm}_n & \rightarrow \text{lm}_c \\
\text{left} & \rightarrow \text{end} \\
\text{lm}_b & \rightarrow \text{lm}_a \\
\text{leave}(\text{lm}_n), \text{gostraight}, \text{turn}(\text{right}, \text{lm}_c), \\
\text{pass}(\text{lm}_b, \text{left}), \text{turn}(\text{left}, \text{lm}_b), \text{observe}(\text{lm}_a, \text{right})
\end{align*}
\]

- A* search
- determine reorientation points
- pick landmarks for reorientation points
- pick landmarks for long straight segments
- map to a sequence of messages
- determine how to refer to landmarks

Today

- Mapping semantics to syntax
- Content determination
- Referring Expression Generation
  - Multimodal referring expressions
    - generating pointing gestures
    - generating iconic gestures

Referring Expressions

- linguistic expressions referring to objects or sets of objects
- NLG has focused on \textit{definite descriptions}: expressions of the form ‘the N’ that uniquely identifies an object in a given context

\[
\text{the rabbit in the hat on the table}
\]

\[
\text{the red rabbit}
\]

\[
\text{the small yellow rabbit}
\]

Generating Definite Descriptions

\textbf{Task}: Find a description that uniquely identifies the target entity.
Lots of different algorithms


Differences:

• expressivity; e.g. in terms of Description Logics:
  
  Dale & Reiter (1995)  CL
  van Deemter (2002a)  PL
  Dale & Haddock (1991)  EL
  Gardent (2002)  ELU\rightarrow
  Krahmer et al. (2003)  EL + nominals (hybrid logic)

• representation of the description, strategy for constructing it, and way of determining success

Dale & Reiter: Incremental Algorithm

Input: a set of individuals with properties
a target entity
Output: a set of properties

Algorithm: start with an empty set

Input: a set of individuals with properties
a target entity
Output: a set of properties
Algorithm: start with an empty set

add properties until the target has no distractors (other individuals that fit the description)
Dale & Reiter: Incremental Algorithm

**Input:** a set of individuals with properties
- a target entity

**Output:** a set of properties

**Algorithm:**
- start with an empty set
- add properties until the target has no distractors (other individuals that fit the description)
- consider properties in this order: type > color > size

- **target:** r4
- **distractors:** \{r1, r2, r3, r5\}
- **properties:** ∅

---

Dale & Haddock (1991): Extensions to relations

**Input:** a set of individuals with properties
- a target entity

**Output:** a set of properties

**Algorithm:**
- start with an empty set
- add properties until the target has no distractors (other individuals that fit the description)
- consider properties in this order: type > color > size

- **target:** r4
- **distractors:** ∅
- **properties:** \{orange\}

---

**General strategy:**

- maintain a stack of targets
- focus on the one at the top
- when adding a relation, push the new individual onto the stack

**Problem:**

- infinite recursion

-the rabbit in the hat containing the rabbit ...
• domain as graph

• general idea: find a subgraph (covering the target) that can only be placed in the domain graph in one way
Krahmer, Erk & Verleg: A graph based algorithm

- domain as graph
  - rabbit
  - in
  - on
  - hat
  - on
  - table
  - on
  - floor

**target**

NOTE: no problem with relations.

Krahmer, Erk & Verleg: the cost of REs

- If there are several possibilities, which one is best?

  ![Possible configurations with costs](image)

  → The one with the lowest cost.

  E.g.: each vertex: 1, type properties: 1, relations: 2
  - many different cost schemes are possible

Krahmer, Erk & Verleg: searching for the cheapest RE

- branch-and-bound search
- Will always find cheapest RE.
- Which solution is found first depends on order in which subgraph is built.
- First solution gives a first upper bound on the cost which needs to be underbid by later solution candidates.

Arecs, Koller & Striegnitz: Description logic formulas as REs
**Background**

- DL formulas denote sets of individuals
- REG = compute a DL formula that denotes exactly the singleton set containing the target
- One DL problem: given a model, find all groups of individuals that cannot be distinguished from each other through the logical language. (similarity sets)
- There are very efficient algorithms for computing similarity sets.
- Our approach: adapt such an algorithm for REG.

**Description Logics**

<table>
<thead>
<tr>
<th>DL formulas</th>
<th>interpretation</th>
<th>example</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\top$</td>
<td>$\Delta$</td>
<td>${r_1, r_2, r_3, h_1, h_2}$</td>
</tr>
<tr>
<td>$\not\varphi$</td>
<td>$\Delta - |\varphi|$</td>
<td>${h_1, h_2}$</td>
</tr>
<tr>
<td>$\varphi \lor \varphi'$</td>
<td>$|\varphi| \lor |\varphi'|$</td>
<td>${r_1, r_2}$</td>
</tr>
<tr>
<td>$\exists R. \varphi$</td>
<td>${i \mid \text{for some } i', i' \in |\varphi| \text{ and } (i, i') \in |R|}$</td>
<td>${r_1, r_3}$</td>
</tr>
</tbody>
</table>

**$\mathcal{L}$-Similarity**

Individual $i$ is $\mathcal{L}$-similar to $i'$ if there is no $\mathcal{L}$-formula that holds of $i$ but not of $i'$.

$r_j$ is EL-similar to $r_x$, but not vice versa.

$r_j$ is not ALC-similar to $r_2$. 

---

Kristina Striegnitz, Union College - ESSLI 2008 Aug 11-15
The $L$-similarity set of $i$ is the set of all individuals to which $i$ is $L$-similar.

For every $L$-similarity set there is an $L$-formula that denotes exactly the individuals in the set.
Algorithm 1: Computing the \( L \)-similarity sets

Input: A model \( M = (\Delta, \cdot, \mathcal{I}) \)

Output: A set \( RE \) of formulas such that \( \{\varphi \mid \varphi \in RE\} \) is the set of \( L \)-similarity sets of \( M \).

1. \( RE = \{T\} \)
2. for \( p \in \text{prop} \) do
3. \( \text{add}_L(p, RE) \)
4. while exists some \( \varphi \in RE, ||\varphi||^M > 1 \) do
5. for \( \varphi \in RE, R \in \text{rel} \) do
6. \( \text{add}_L(\exists R \varphi, RE) \)
7. if made no changes to \( RE \) then
8. exit

Algorithm 3: \( \text{add}_L(\varphi, RE) \)

1. for \( \psi \in RE \) with \( ||\varphi|| > 1 \) do
2. if \( \psi \sqcap \varphi \) is not subsumed in \( RE \) and
3. \( ||\psi \sqcap \varphi|| \neq 0 \) and \( ||\psi \sqcup \varphi|| \neq ||\psi|| \) then
4. \( \text{add} \psi \sqcap \varphi \) to \( RE \)
5. remove subsumed formulas from \( RE \)

Algorithm 3: \( \text{add}_L(\varphi, RE) \)

1. for \( \psi \in RE \) with \( ||\varphi|| > 1 \) do
2. if \( \psi \sqcap \varphi \) is not subsumed in \( RE \) and
3. \( ||\psi \sqcap \varphi|| \neq 0 \) and \( ||\psi \sqcup \varphi|| \neq ||\psi|| \) then
4. \( \text{add} \psi \sqcap \varphi \) to \( RE \)
5. remove subsumed formulas from \( RE \)

Features

- Res for all individuals are computed in parallel
- Very efficient
- Order of properties is the only way to control the resulting description
- Lots of possible extensions using existing DL algorithms and results

Today

- Mapping semantics to syntax
- Content determination
- Referring Expression Generation
  - Multimodal referring expressions
    - Generating pointing gestures
    - Generating iconic gestures
Multimodal referring expressions

Examples

Generating pointing gestures

- when to point
- how precisely to point (to object or to region)
- what info to put into the accompanying language

Some work on generating pointing gestures:
Claasen (1992)
Lester et al. (1999)
Kranstedt & Wachsmuth (2005)
vander Sluis & Krahmer (2007) ← extends the graph-based algorithm

Lester et al. 1999

- Can the referent easily be confused with other objects?
  
  Are there recently mentioned objects nearby?
  
  Are there other objects of the same type nearby?

  If so, point.

- Pointing is always unambiguous. If necessary, the agent moves toward the object to point.

van der Sluis & Krahmer: REs + pointing gestures

- different levels of pointing are represented as labels in the domain graph

- what level of pointing and what linguistic material is chosen depends on the costs

- cost of pointing depends on size of target and on the distance the hand has to move:

\[
\text{cost(pointing)} = \log_D(D/W + 1)
\]
Generating iconic gestures

- when to use iconic gesture
- what gesture to use

When to use iconic gestures

- with rhetorical material (roughly: material that contributes new information to the discourse) (Cassell 2000)
- dependent on domain:
  - to express shape and location in object descriptions (Yan 2000)
  - to express path, manner and speed in motion descriptions (Cassell & Prevost 1996)

What gesture to use

- most commonly: use a gesticon
  - a collection of pre-animated gestures associated with specific semantic meanings
- alternatively: generate gestures on the fly (based on geometric and visible properties of the referent)

NUMACK: Generating gestures on the fly

Goal: Generate gestures on the fly based on information about the referent

Domain: Giving walking directions (across Northwestern University's campus)
  - gestures referring to landmarks
**Iconic Gestures**

- Iconic gestures visually resemble what they depict.
- They encode information that may be redundant with the content of the accompanying speech or may add to it.
- No stable form-meaning pairing:
  - same gesture can be used to refer to different things
  - same thing can be referred to using different gestures
  - gesture on its own is insufficient for interpretation
- Iconic gestures are interpreted in context (speech, previous discourse, domain, dialogue situation) to depict specific entities.

  "it's got [like steeples]"

  "there's a church"

**Gestures referring to landmarks**

- Functions:
  - locating landmarks
  - depicting shape of landmarks
- Many gestures have both a locating and a shape depicting component.
- Speakers take on different perspectives when describing routes.

  "on your left once you hit this parking lot [is the Allen Center]"

  "it's got [like steeples]"

**Route perspective gestures**

- Direction giver takes on perspective of a person walking the route.
- Gestures locate landmarks with respect to this imaginary direction follower’s position and orientation.
- Most common type of gesture for referring to landmarks: 54%.

  "on your left once you hit this parking lot [is the Allen Center]"

**Survey perspective gestures**

- Gestures lay out a map in front of the speaker’s body.
- Landmarks are located with respect to the imaginary direction follower’s body and relative to other landmarks.
- 16% of all gestures referring to landmarks.

  "[University Hall] is on your right, [on the left is Kresge], and [then straight ahead is Harris]"
Non-locating gestures

- do not locate landmarks
- depict shape
- 16% of all gestures referring to landmarks.

"on your left once you hit this parking lot [is the Allen Center]..."

"...and [it’s really big]"

Questions

- When should we use which perspective?
- How is location and shape information depicted in the gesture?

Gesture perspective in the data

Gesture perspective seems to be (at least partly) determined by dialogue function.

- Non-locating gestures tend to occur in elaborations.
- Survey perspective gestures tend to occur in answers to clarification questions and in re-descriptions of route segments.
- Non-locating and survey perspective gestures tend to not occur in plain forward looking statements.
- Route perspective gestures tend to occur in plain statements.
- They tend to not occur in answers to clarification questions, re-descriptions of route segments, or elaborations.

Gesture perspective in the system

- Non-locating gestures are used in elaborations which don't mention the location of the landmark.
  
  E.g.: "Dearborn Observatory is on your left. It is a building with a dome."

- Survey perspective gestures are used for re-descriptions of route segments at “difficult” reorientation points.
- Route perspective is used for all other gestures.
Expressing location in iconic gestures

- Given the position and orientation that a person walking the route would have at the current point of the directions, calculate the angle to the referent(s).
- Map those angles to positions in the gesture space.

route perspective

survey perspective

Some evidence in the data

- gestures in 10 direction giving dialogues coded for gesture morphology (hand shape, hand position, palm direction, extended finger direction)
- landmarks these gestures refer to were coded for salient visual features
- looked at flat handshapes
- hypotheses:
  1) palm down ↔ horizontal surface
  2) fingers up ↔ vertical surface
  3) fingers forward & palm sideways ↔ path
- confirmed hypothesis 2 and 3

Expressing shape in iconic gestures

Hypothesis: gesture morphology is related to visual and spatial properties of the referent.

Problems with the study

- landmarks may have more than one visually salient feature
  - did not take into account discourse context
  - did not take into account direction from which landmark was approached
  - did not differentiate between perspectives
Gesture Planning

**Referent Landmark**

- **Shape**: plane
- **Orientation**: vertical & orthogonal to DF’s orient
- **Primary Axis**: vertical
- **Location wrt. DF**: -10°
- **Perspective**: route

**Salient Geometric Form**

**Image Description Features**

**Gesture Form Features**

**Handshape**: flat (ASL B or 5)
- **Orientation**: vertical & orthogonal to DF’s orient
- **Primary Axis**: vertical
- **Location wrt. DF**: -10°
- **Perspective**: route

**Gesture Planning and Utterance Construction**

**Goal**: describe event e1

- **Shape**: 5 (ASL)
- **Palm Dir.**: away
- **Finger Dir.**: up
- **Trajectory**: linear up
- **Location**: center

**Assert:**
- `rel_loc(e1, b1, df, front)`
- `building(b1)`
- `tall(b1)`
Today

• Mapping semantics to syntax
• Content determination
• Referring Expression Generation
• Multimodal referring expressions
  – generating pointing gestures
  – generating iconic gestures

Tomorrow: discourse and dialogue phenomena