Perceptual Types

Staffan Larsson
University of Gothenburg
Sweden

June 23, 2011

Perceptual Types

Staffan Larsson
University of Gothenburg
Sweden

June 23, 2011

Outline

1. Introduction & the left-or-right game
2. Dynamic semantics in TTR
3. The TTR perceptron
4. Contextual interpretation in the left-or-right game
   - The meaning of "right"
   - Contextual interpretation of "right"
5. Learning perceptual meaning from interaction
   - Detecting foreground inconsistency
   - Updating perceptual meaning
6. Conclusion and future work

Outline

- In dynamic semantics, meanings are context-update functions which take an input context and return an updated (output) context.
- We will present a dynamic semantic approach to subsymbolic perceptual aspects of meaning, using a TTR version of the Perceptron
- This shows how subsymbolic and perceptual aspects of meaning can be incorporated with traditional formal semantics
- Furthermore, we show how subsymbolic aspects of meaning can be updated as a result of observing language use in interaction, thereby enabling fine-grained semantic plasticity and semantic coordination.

Semantic coordination

- Research on alignment shows that agents negotiate domain-specific microlanguages for the purposes of discussing the particular domain at hand
  - (Clark and Wilkes-Gibbs, 1986; Garrod and Anderson, 1987; Pickering and Garrod, 2004; Brennan and Clark, 1996; Healey, 1997; Larsson, 2007)
- Two agents do not need to share exactly the same linguistic resources (grammar, lexicon etc.) in order to be able to communicate
- An agent's linguistic resources can change during the course of a dialogue when she is confronted with a (for her) innovative use
- *Semantic coordination*: the process of interactively coordinating the meanings of linguistic expressions.

Mechanisms for semantic coordination in dialogue

Some mechanisms for semantic coordination in dialogue:

- Corrective feedback, where one DP implicitly corrects the way an expression is used by another DP
- Explicit definitions and negotiations of meanings
- **“Silent” coordination, by DPs observing the language use of others and adapting to it**

The left-or-right game

- A and B are facing a framed surface on a wall, and A has a bag of objects which can be attached to the framed surface.
  1. A places an object in the frame
  2. B orients to the new object, assigns it a unique individual marker and labels it "foc-obj" in B's take on the situation
  3. A says either "left" or "right"
  4. B interprets A's utterance based on B's take on the situation. Interpretation involves determining whether B's understanding of A's utterance is consistent with B's take on the situation.
  5. If an inconsistency results from interpretation, B assumes A is right, says "aha", and learns from this exchange; otherwise, B says "okay"
- There are many possible variants of this game, which will be explored in future research.

- The left-or-right game can be regarded as a considerably pared-down version of the “guessing game” in Steels and Belpaeme (2005), where perceptually grounded colour terms are learnt from interaction.
- The kinds of meanings learnt in the left-or-right game may be considered trivial.
- However, at the moment we are mainly interested in the basic principles of combining formal dynamic semantics with learning of perceptual meaning from dialogue
- The hope is that these can be formulated in a general way which can later be used in more interesting settings.
- Most of the remainder of this presentation will be spent formulating this simple game in TTR.

Outline

Records in TTR

- Record on the left, record type on the right:

$$\begin{bmatrix} l_1 & = & a_1 \\ l_2 & = & a_2 \\ \ldots & & \\ l_n & = & a_n \\ \ldots & & \end{bmatrix} : \begin{bmatrix} l_1 & : & T_1 \\ l_2 & : & T_2(l_1) \\ \ldots & & \\ l_n & : & T_n(l_1, l_2, \ldots, l_{n-1}) \end{bmatrix}$$

- A record of the type to the right has to have fields with the same labels as those in the type
- It may also include additional fields not required by the type
- In place of the types which occur to the right of ':' in the record type, the record must contain an object of that type.

Predicates and proofs

- Sample record and record type

$$\left[\begin{array}{lcl} \text{ref} & = & \text{obj}_{123} \\ \text{c}_{panda} & = & \text{prf}(\text{panda}(\text{obj}_{123})) \end{array}\right] : \left[\begin{array}{lcl} \text{ref} & : & \text{Ind} \\ \text{c}_{panda} & : & \text{panda}(\text{ref}) \end{array}\right]$$

- Types can be constructed from predicates (panda); these correspond to propositions in FOL
- Objects of types constructed from predicates are proofs; "Propositions are types of proofs"

Dependent types

- Sample record type

$$\begin{bmatrix} \text{ref} & : & \text{Ind} \\ c_{panda} & : & \text{panda(ref)} \end{bmatrix}$$

- Types constructed with predicates may also be *dependent*
- This is represented by the fact that arguments to the predicate may be represented by labels used on the left of the ':' elsewhere in the record type.

Manifest fields

- Some of our types will contain *manifest fields* (Coquand *et al.*, 2004) like the ref-field in the following type:

$$\left[\begin{array}{lll} \text{ref}{=}\text{obj}_{123} & : & \text{Ind} \\ \text{c}_{panda} & : & \text{panda(ref)} \end{array}\right]$$

- $\left[\text{ref}{=}\text{obj}_{123}{:}\text{Ind}\right]$ is a convenient notation for $\left[\text{ref} : \text{Ind}_{\mathrm{obj123}}\right]$ where $\text{Ind}_{\mathrm{obj123}}$ is a *singleton type*
- If $a : T$, then T_a is a singleton type and $b : T_a$ (i.e. b is of type T_a) iff $b = a$.

Subtyping

An important notion in this kind of type theory is that of *subtype*, represented by $\sqsubseteq$:

$$\left[\begin{array}{lll} \text{ref} & : & \text{Ind} \\ \text{c}_{panda} & : & \text{panda(ref)} \end{array}\right] \sqsubseteq \left[\begin{array}{lll} \text{ref} & : & \text{Ind} \end{array}\right]$$

$$\left[\begin{array}{lll} \text{ref}{=}\text{obj}_{123} & : & \text{Ind} \end{array}\right] \sqsubseteq \left[\begin{array}{lll} \text{ref} & : & \text{Ind} \end{array}\right]$$

Merging record types (unification)

If

$$T_1 = \begin{bmatrix} \mathrm{x} : & \mathit{Ind} \\ \mathrm{c_{panda}} : & \mathrm{panda(x)} \end{bmatrix}$$

and

$$T_2 = \begin{bmatrix} \mathrm{x} : & \mathit{Ind} \\ \mathrm{c_{animal}} : & \mathrm{animal(x)} \end{bmatrix}$$

then

$$T_1 \underset{\cdot}{\wedge} T_2 = \begin{bmatrix} \mathrm{x} : & \mathit{Ind} \\ \mathrm{c_{panda}} : & \mathrm{panda(x)} \\ \mathrm{c_{animal}} : & \mathrm{animal(x)} \end{bmatrix}$$

Background and foreground meaning

- We will take parts of the meaning of an uttered expression to be *foregrounded*, and other parts to be *backgrounded*.
- Background meaning represents constraints on the context, whereas foreground material is the content - the information to be added to the context by the utterance.
- Both background and foreground meaning components are represented in TTR as record types T_{bg} and T_{fg} (where the latter may depend on the former)

Static meaning

- A static meaning (Kaplans “character”) can be seen as a functions from context to content
- In TTR, static meanings are functions from records (representing contexts) to record types (representing contents).

$$\lambda r : T_{bg}(T_{fg})$$

- “a man runs”

$$\lambda r : \begin{bmatrix} \text{ref} & : & \text{Ind} \\ \text{c}_{man} & : & \text{man(ref)} \end{bmatrix} \left(\begin{bmatrix} \text{c}_{run} & : & \text{run(t.ref)} \end{bmatrix} \right)$$

Dynamic meanings as updates to takes on contexts

- We take updates in dynamic semantics to be updates of takes on contexts.
- In TTR dynamic semantics, we therefore need a function which takes an agents take on the context as input and returns an updated take on the context. In TTR terms, this means we need a function from record types to record types.

$$\lambda t \sqsubseteq T_{bg}(t \dot{\wedge} T_{fg})$$

- "a man runs"

$$\lambda t \sqsubseteq \begin{bmatrix} \text{ref} & : & \text{Ind} \\ \text{c}_{man} & : & \text{man(ref)} \end{bmatrix} (\text{t} \dot{\wedge} \begin{bmatrix} \text{c}_{run} & : & \text{run(r.ref)} \end{bmatrix})$$

Meanings in the lexicon

- When representing the meaning of an expression e in lexicon, we can use a record collecting the various aspects of $[e]$, the meaning of e:

$$[\mathrm{e}] = \begin{bmatrix} \mathrm{bg} & = & T_{bg} \\ \mathrm{fg} & = & T_{fg} \\ \mathrm{sfun} & = & \lambda r : \mathrm{bg}(\mathrm{fg}[\mathrm{bg}/r]) \\ \mathrm{dfun} & = & \lambda t \sqsubseteq \mathrm{bg}(x \dot{\wedge} \mathrm{fg}[\mathrm{bg}/t]) \end{bmatrix}$$

- where $e_1[e_2/e_3]$ is e_1 with any occurrences of e_2 replaced by e_3.

- "a man runs"

$$[\mathrm{e}] = \begin{bmatrix} \mathrm{bg} & = & \begin{bmatrix} \mathrm{ref} & : & \mathrm{Ind} \\ \mathrm{c}_{man} & : & \mathrm{man}(\mathrm{ref}) \end{bmatrix} \\ \mathrm{fg} & = & \begin{bmatrix} \mathrm{c}_{run} & : & \mathrm{run}(\mathrm{bg.ref}) \end{bmatrix} \\ \mathrm{sfun} & = & \lambda r : \mathrm{bg}(\mathrm{fg}[\mathrm{bg}/r]) \\ \mathrm{dfun} & = & \lambda t \sqsubseteq \mathrm{bg}(t \dot{\wedge} \mathrm{fg}[\mathrm{bg}/t]) \end{bmatrix}$$

Takes on situations

- In TTR, contexts/situations are represented as records, whereas an agent's *takes* on a context is represented as a record type (typically involving manifest fields).
- This allows takes on contexts to be underspecified, which is useful in modeling agents with incomplete knowledge.
- An agent's take on a situation is a part of the agent's information state.

$$\begin{bmatrix} \text{ref}=\text{a} & : & \text{Ind} \\ \text{c}_{man} & : & \text{man(ref)} \end{bmatrix}$$

Contextual interpretation using dynamic meaning

$$[\text{e}].\text{dfun} = \lambda t \sqsubseteq \text{bg}(t \dot{\wedge} \text{fg}[\text{bg}/t])$$

- Dynamic contextual interpretation amounts to applying this function to the input (take on the) context, and the result of function application is the output (take on the) context

$$[\text{a man runs}].\text{dfun}@\begin{bmatrix} \text{ref=a} & : & \text{Ind} \\ c_{man} & : & \text{man(ref)} \end{bmatrix} =$$
$$\begin{bmatrix} \text{ref=a} & : & \text{Ind} \\ c_{man} & : & \text{man(ref)} \\ c_{run} & : & \text{run(ref)} \end{bmatrix}$$

Foreground inconsistency

- Given our formulation of the context update function [e].dfun corresponding to a meaning [e], one thing that might go wrong when applying it to an input context s is that the output context is inconsistent[1]:
  - $[e].\text{dfun}@(s) \approx \bot$
- This is a case of *foreground inconsistency*
- A record type RT which contains both a field with type T and a field with type $\neg T$ is inconsistent ($RT \approx \bot$).
- If a record type is inconsistent, there can be no records of that type; it is not only uninhabited but uninhabitable.

[1] We use @ to denote function application.

Outline

Perceptual meaning

- The lexical meaning [*e*] of an expression *e* contains not only compositional and ontological semantics but also perceptual meaning.
- Perceptual meaning = that aspect of the meaning of an expression which allows an agent to detect objects or situations referred to by the expression *e*.
  - Knowing the perceptual meaning of "panda" allows an agent to correctly classify pandas in her environment as pandas.
  - An agent which is able to compute the perceptual meaning of "a boy hugs a dog" will be able to correctly classify situations where a boy hugs a dog.
- We can think of perceptual meanings as classifiers of sensory input
- Example: the TTR perceptron

Perceptrons

- Classification of perceptual input can be regarded as a mapping of sensor readings to types
- To represent perceptual classifiers, we will be using a simple perceptron - a very simple neuron-like object with several inputs and one output.

$$o(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{w} \cdot \mathbf{x} > \text{t} \\ 0 & \text{otherwise} \end{cases}$$

  where $\mathbf{w} \cdot \mathbf{x} = \sum_{i=1}^{n} w_i x_i = w_1 x_1 + w_2 x_2 + \ldots + w_n x_n$

- Limited to learning problems which are linearly separable; the distinction between left and right is one such problem.

Vectors in TTR

- In TTR, an n-dimensional real-valued vector will be represented as a record with labels 1, ..., n where the value of each label will be a real number.
- Such a records will be of the type RealVector_n.

$$\text{RealVector}_n = \begin{bmatrix} 1 & : & \text{Real} \\ 2 & : & \text{Real} \\ \ldots & & \\ \text{n} & : & \text{Real} \end{bmatrix}$$

$$\mathbf{x} = \begin{bmatrix} 1 & = & 0.23 \\ 2 & = & 0.34 \\ 3 & = & 0.45 \end{bmatrix} : \text{RealVector}_3$$

For convenience, we will abbreviate this as in this example:

$$\mathbf{x} = \begin{bmatrix} 0.23 & 0.34 & 0.45 \end{bmatrix} : \text{RealVector}_3$$

$\mathbf{x}_n = \mathbf{x}.\text{n}$, so $\mathbf{x}_2 = 0.34$

The TTR perceptron cont'd

A TTR perceptron can be represented as a record:

$$\left[\begin{array}{lll} \mathrm{w} & = & \begin{bmatrix} 0.800 & 0.010 \end{bmatrix} \\ \mathrm{t} & = & 0.090 \\ \mathrm{fun} & = & \lambda v : \mathrm{RealVector} \\ & & \quad \left(\begin{cases} 1 & \text{if } v \cdot \mathrm{w} > \mathrm{t} \\ 0 & \text{otherwise} \end{cases} \right) \end{array}\right]$$

Where p.fun will evaluate to

$$\lambda v : \mathrm{RealVector} \left(\begin{cases} 1 & \text{if } v \cdot \begin{bmatrix} 0.800 & 0.010 \end{bmatrix} > 0.090 \\ 0 & \text{otherwise} \end{cases} \right)$$

- This representation allows modifying w and t by updating the record

The TTR classifier perceptron

- The basic perceptron returns a real-valued number (1 or 0) but when we use a perceptron as a classifier we want it to instead return a type.
- Typically, such types will be built from a predicate and some number of arguments; a type of proof, or a "proposition".

A TTR classifier perceptron for a type P can be represented as a record:

$$\left[\begin{array}{lll} \mathrm{w} & = & \begin{bmatrix} 0.800 & 0.010 \end{bmatrix} \\ \mathrm{t} & = & 0.090 \\ \mathrm{fun} & = & \lambda v : \mathrm{RealVector} \\ & & \quad\left(\begin{cases} P & \text{if } v \cdot \mathrm{w} > \mathrm{t} \\ \neg P & \text{otherwise} \end{cases} \right) \end{array}\right]$$

Outline

Sensors and the focused object

- In first language acquisition, training of perceptual meanings typically takes place in situations where the referent is in the shared focus of attention and thus perceivable to the dialogue participants
- For the time being we limit our analysis to such cases.
- We assume that our DPs are able to establish a shared focus of attention
- A (simple) sensor collects some information (sensor input) from the environment and emits a real-valued vector.
- The sensor is assumed to be oriented towards the object in shared focus of attention.

Example

B's take on the situation prior to playing the left-or-right game:

$$s_1^B = \begin{bmatrix} \text{sr}_{pos} = \begin{bmatrix} 0.900 & 0.100 \end{bmatrix} & : & \text{RealVector} \\ \text{foc-obj} = \text{obj}_{45} & : & \text{Ind} \\ \text{spkr} = \text{A} & : & \text{Ind} \end{bmatrix}$$

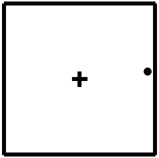
- In the left-or-right game, we will assume that B's take on the situation includes
  - a reading from a position sensor (denoted "sr_{pos}")
  - a field foc-obj for an object in shared focus of attention.
- The position sensor returns a two-dimensional real-valued vector representing the horizontal vertical coordinates of the focused object:
  - $\begin{bmatrix} x & y \end{bmatrix}$ where $-1.0 \leq x, y \leq 1.0$ and $\begin{bmatrix} 0.0 & 0.0 \end{bmatrix}$ represents the center of the frame.
- In s_1^B, B's sensor is oriented towards obj_{45} and sr_{pos} returns a vector corresponding to the position of obj_{45}.

Sensors readings as proofs

- “Propositions are types of proofs”
- One can have different ideas of what kind of objects count as proofs
- Here we will be assuming that proof-objects can be *takes on situations* involving *readings from sensors*; we can call such a proof a *verification*
- Types of verifications are “Perceptual types” or “Verification types”

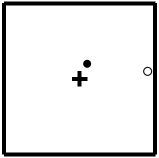
A sample interaction

A: "right"



B: "okay"

A: "right"



B: "aha"

The meaning of “right”

- We can now say what a meaning in B’s lexicon might look like before a round of the left-or-right game.
- We assume that B has meanings only for “left” and “right”, which could be explicated as “the object in focus is to the left” and “the object in focus is to the right”.
- In our representations of meanings, we will combine the TTR representations of meanings with the TTR representation of classifier perceptrons.

Agent B's initial take on the meaning of "right" , $[\text{right}]^B =$

$$
\left[\begin{array}{l}
\text{w} = \begin{bmatrix}0.800 & 0.010\end{bmatrix} \\
\text{t} = 0.090 \\
\text{bg} = \left[\begin{array}{lcl} \text{sr}_{pos} & : & \text{RealVector} \\ \text{foc-obj} & : & \text{Ind} \\ \text{spkr} & : & \text{Ind} \end{array}\right] \\
\text{fg} = \left[\begin{array}{l}
\text{c}^{perc}_{right} = \left[\begin{array}{l} \text{sr}_{pos} = \text{bg.sr}_{pos} \\ \text{foc-obj} = \text{bg.foc-obj} \end{array}\right] : \\
\qquad \begin{cases} \text{right(bg.foc-obj)} & \text{if bg.sr}_{pos}\cdot\text{w} > \text{t} \\ \neg\text{right(bg.foc-obj)} & \text{otherwise} \end{cases} \\
\text{c}^{tell}_{right} = \left[\begin{array}{l} \text{str} = \text{"right"} \\ \text{spkr} = \text{bg.spkr} \\ \text{foc-obj} = \text{bg.foc-obj} \end{array}\right] : \text{right(bg.foc-obj)}
\end{array}\right] \\
\text{sfun} = \lambda x : \text{bg}(\text{fg}[\text{bg}/x]) \\
\text{dfun} = \lambda x \sqsubseteq \text{bg}(x \dot{\wedge} \text{fg}[\text{bg}/x])
\end{array}\right]
$$

[right].w, t and bg fields

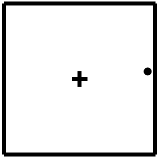
- The fields w and t specify weights and a threshold for a classifier perceptron which is used to classify sensor readings.
- The bg field represents constraints on the input context, which requires that there is a colour sensor reading and a focused object foc-obj.

Contextual interpretation of "right"

- We will first show a case where interpretation runs smoothly.
- Player A picks up an object and places it in the frame, and B finds the object and assigns it the individual marker obj_{45}, directs the position sensor to it and gets a reading.
- Player A now says "right", after which B's take on the situation is $^{B}_{1}$, repeated here for convenience:

$$s_1^B = \begin{bmatrix} \mathrm{sr}_{pos} = \begin{bmatrix} 0.900 & 0.100 \end{bmatrix} & : & \mathrm{RealVector} \\ \text{foc-obj} = \mathrm{obj}_{45} & : & \mathrm{Ind} \\ \mathrm{spkr} = \mathrm{A} & : & \mathrm{Ind} \end{bmatrix}$$

A: “right”



To interpret A's utterance, B applies $[\text{right}]^B$.dfun to s_1^B to yield a new take on the situation s_2^B:

$$s_2^B = [\text{right}]^B.\text{dfun}@s_1^B = \begin{bmatrix} \text{sr}_{pos} = \begin{bmatrix} 0.900 & 0.100 \end{bmatrix} : \text{RealVector} \\ \text{foc-obj} = \text{obj}_{45} : \text{Ind} \\ \text{spkr} = \text{A} : \text{Ind} \\ \text{c}^{perc}_{right} = \begin{bmatrix} \text{sr}_{pos} = \begin{bmatrix} 0.900 & 0.100 \end{bmatrix} \\ \text{foc-obj} = \text{obj}_{45} \end{bmatrix} : \text{right}(\text{obj}_{45}) \\ \text{c}^{tell}_{right} = \begin{bmatrix} \text{str} = \text{``right''} \\ \text{spkr} = \text{A} \\ \text{foc-obj} = \text{obj}_{45} \end{bmatrix} : \text{right}(\text{obj}_{45}) \end{bmatrix}$$

$$s_2^B = [\text{right}]^B.\text{dfun}@s_1^B =$$

$$\lambda x \sqsubseteq \begin{bmatrix} \text{sr}_{pos} & : & \text{RealVector} \\ \text{foc-obj} & : & \text{Ind} \\ \text{spkr} & : & \text{Ind} \end{bmatrix}$$

$$(x \dot{\wedge} \begin{bmatrix} c_{right}^{perc} = \begin{bmatrix} \text{sr}_{pos} = x.\text{sr}_{pos} \\ \text{foc-obj} = x.\text{foc-obj} \end{bmatrix} : \begin{cases} \text{right}(x.\text{foc-obj}) & \text{if } x.\text{sr}_{pos} \cdot \begin{bmatrix} 0.800 & 0.010 \end{bmatrix} > 0.090 \\ \neg\text{right}(x.\text{foc-obj}) & \text{otherwise} \end{cases} \\ c_{right}^{tell} = \begin{bmatrix} \text{str} = \text{“right”} \\ \text{spkr} = x.\text{spkr} \\ \text{foc-obj} = x.\text{foc-obj} \end{bmatrix} : \text{right}(x.\text{foc-obj}) \end{bmatrix})$$

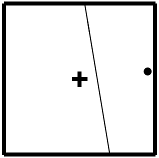
$$@$$

$$\begin{bmatrix} \text{sr}_{pos} = \begin{bmatrix} 0.900 & 0.100 \end{bmatrix} & : & \text{RealVector} \\ \text{foc-obj} = \text{obj}_{45} & : & \text{Ind} \\ \text{spkr} = \text{A} & : & \text{Ind} \end{bmatrix}$$

$$\left[\begin{array}{l}
\mathrm{sr}_{pos}=\begin{bmatrix}0.900 & 0.100\end{bmatrix}:\mathrm{RealVector}\\
\text{foc-obj}=\mathrm{obj}_{45}:\mathrm{Ind}\\
\mathrm{spkr}=\mathrm{A}:\mathrm{Ind}\\
\mathrm{c}^{perc}_{right} = \begin{bmatrix}\mathrm{sr}_{pos} = \begin{bmatrix}0.900 & 0.100\end{bmatrix}\\ \text{foc-obj} = \mathrm{obj}_{45}\end{bmatrix} : \\
\qquad \begin{cases}\mathrm{right}(\mathrm{obj}_{45}) & \text{if } \begin{bmatrix}0.900 & 0.100\end{bmatrix}\cdot\begin{bmatrix}0.800 & 0.010\end{bmatrix} > 0.090\\ \neg\mathrm{right}(\mathrm{obj}_{45}) & \text{otherwise}\end{cases}\\
\mathrm{c}^{tell}_{right}=\begin{bmatrix}\mathrm{str} = \text{“right”}\\ \mathrm{spkr} = \mathrm{A}\\ \text{foc-obj} = \mathrm{obj}_{45}\end{bmatrix} : \mathrm{right}(\mathrm{obj}_{45})
\end{array}\right]$$

$$\left[\begin{array}{ll}
\text{sr}_{pos} = \begin{bmatrix} 0.900 & 0.100 \end{bmatrix} : \text{RealVector} & \\
\text{foc-obj} = \text{obj}_{45} : \text{Ind} & \\
\text{spkr} = \text{A} : \text{Ind} & \\
\text{c}^{perc}_{right} = \begin{bmatrix} \text{sr}_{pos} = \begin{bmatrix} 0.900 & 0.100 \end{bmatrix} \\ \text{foc-obj} = \text{obj}_{45} \end{bmatrix} & : \text{right}(\text{obj}_{45}) \\
\text{c}^{tell}_{right} = \begin{bmatrix} \text{str} = \text{“right”} \\ \text{spkr} = \text{A} \\ \text{foc-obj} = \text{obj}_{45} \end{bmatrix} & : \text{right}(\text{obj}_{45})
\end{array}\right]$$

A: “right”



B: “okay”

Proofs of right(obj_{45})

$$\begin{bmatrix} sr_{pos} = \begin{bmatrix} 0.900 & 0.100 \end{bmatrix} : RealVector \\ foc\text{-}obj = obj_{45} : Ind \\ spkr = A : Ind \\ c^{perc}_{right} = \begin{bmatrix} sr_{pos} = \begin{bmatrix} 0.900 & 0.100 \end{bmatrix} \\ foc\text{-}obj = obj_{45} \end{bmatrix} : right(obj_{45}) \\ c^{tell}_{right} = \begin{bmatrix} str = \text{``right''} \\ spkr = A \\ foc\text{-}obj = obj_{45} \end{bmatrix} : right(obj_{45}) \end{bmatrix}$$

- The value of c^{perc}_{right} is a proof of right(obj_{45})
  - The situation (type) acting as a verification of right(obj_{45}) can be thought of as a "snapshot" of relevant parts of the situation, consisting of the current sensor reading and a specification of the currently focused object.

A further proof of right(obj_{45})

$$\begin{bmatrix} sr_{pos} = \begin{bmatrix} 0.900 & 0.100 \end{bmatrix} : RealVector \\ foc\text{-}obj = obj_{45} : Ind \\ spkr = A : Ind \\ c^{perc}_{right} = \begin{bmatrix} sr_{pos} & = & \begin{bmatrix} 0.900 & 0.100 \end{bmatrix} \\ foc\text{-}obj & = & obj_{45} \end{bmatrix} : right(obj_{45}) \\ c^{tell}_{right} = \begin{bmatrix} str & = & \text{“right”} \\ spkr & = & A \\ foc\text{-}obj & = & obj_{45} \end{bmatrix} : right(obj_{45}) \end{bmatrix}$$

- The value of c^{tell}_{right} is a record containing information about an utterance, namely that A uttered the word “right” when foc-obj was in focus.
  - We assume that this counts as a proof that foc-obj is (to the) right.
  - This implements an assumption that A is always right, an assumption that one could choose to remove in a more complicated version of the left-or-right game.

Aside: Verifications as justifications

- A verification such as $\begin{bmatrix} \text{sr}_{pos} = \begin{bmatrix} 0.900 & 0.100 \end{bmatrix} \\ \text{foc-obj} = \text{obj}_{45} \end{bmatrix}$ could be used in a discussion of $\text{right}(\text{obj}_{45})$
- When faced with the question "How do you know that it [pointing to obj_{45}] is to the right?", A could answer
  - "When I direct my location sensor to it, my right-classifier outputs a 1", or in more colloquial terms,
  - "I can see that it is to the right"

Outline

Learning perceptual meaning from interaction

We now assume that in the next round, A places another object in a different position in the frame and again says "right".
Now, B's take on the situation is as follows:

$$s_3^B = \begin{bmatrix} \text{sr}_{pos} = \begin{bmatrix} 0.100 & 0.200 \end{bmatrix} : \text{RealVector} \\ \text{foc-obj} = \text{obj}_{46} : \text{Ind} \\ \text{spkr} = \text{A} : \text{Ind} \\ \text{c}_{right}^{perc} = \begin{bmatrix} \text{sr}_{pos} = \begin{bmatrix} 0.900 & 0.100 \end{bmatrix} \\ \text{foc-obj} = \text{obj}_{45} \end{bmatrix} : \text{right}(\text{obj}_{45}) \\ \text{c}_{right}^{tell} = \begin{bmatrix} \text{str} = \text{"right"} \\ \text{spkr} = \text{A} \\ \text{foc-obj} = \text{obj}_{45} \end{bmatrix} : \text{right}(\text{obj}_{45}) \end{bmatrix}$$

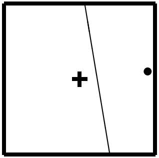
Note that foc-obj has been updated and that there is a new sensor reading.

As before, B interprets A's utterance to yield a new take on the situation:

$$
\mathrm{s}_4^B = [\mathrm{right}]^B.\mathrm{dyn}@\mathrm{s}_3^B =
$$

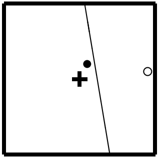
$$
\begin{bmatrix}
\mathrm{sr}_{pos} = \begin{bmatrix} 0.100 & 0.200 \end{bmatrix} : \mathrm{RealVector} \\
\text{foc-obj} = \mathrm{obj}_{46} : \mathrm{Ind} \\
\mathrm{spkr} = \mathrm{A} : \mathrm{Ind} \\
\mathrm{c}^{perc}_{right} = \begin{bmatrix} \mathrm{sr}_{pos} = \begin{bmatrix} 0.900 & 0.100 \end{bmatrix} \\ \text{foc-obj} = \mathrm{obj}_{45} \end{bmatrix} : \mathrm{right}(\mathrm{obj}_{45}) \\
\mathrm{c}^{tell}_{right} = \begin{bmatrix} \mathrm{str} = \text{“right”} \\ \mathrm{spkr} = \mathrm{A} \\ \text{foc-obj} = \mathrm{obj}_{45} \end{bmatrix} : \mathrm{right}(\mathrm{obj}_{45}) \\
\mathrm{c1}^{perc}_{right} = \begin{bmatrix} \mathrm{sr}_{pos} = \begin{bmatrix} 0.100 & 0.200 \end{bmatrix} \\ \text{foc-obj} = \mathrm{obj}_{46} \end{bmatrix} : \neg\ \mathrm{right}(\mathrm{obj}_{46}) \\
\mathrm{c1}^{tell}_{right} = \begin{bmatrix} \mathrm{str} = \text{“right”} \\ \mathrm{spkr} = \mathrm{A} \\ \text{foc-obj} = \mathrm{obj}_{46} \end{bmatrix} : \mathrm{right}(\mathrm{obj}_{46})
\end{bmatrix}
$$

A: “right”



B: “okay”

A: “right”



- This time, however, applying the classifier perceptron to the sensor input yields ¬right(obj_{46})
- Hence the classifier takes s_3^B to contain a proof both of ¬right(obj_{46}) (labelled $c1_{right}^{perc}$) and of right(obj_{45}) (labelled $c1_{right}^{tell}$).
- This is a case of foreground inconsistency – the record type s_4^B is inconsistent ($s_4^B \approx \perp$).
- That is, there can be no situation (record) of this type.
- According to the rules of the game, B resolves this conflict by trusting A's judgement over B's own classification.
- Hence, B must remove $c1_{right}^{perc}$. Furthermore, B can learn from this exchange by updating the weights used by the classifier perceptron associated with [right].

Updating perceptual meaning

Perceptrons are updated using the *perceptron training rule*:

$$w_i \leftarrow w_i + \Delta w_i$$

where

$$\Delta w_i = \eta(o_t - \mathrm{o})x_i$$

where o_t is the target output, o is the actual output, and w_i is associated with input x_i.

- Note that if $o_t = o$, there is no learning.
- However, since B only tries to learn from mistakes (and not from successes), we have already established that $o_t - o$ is 1.0 for a perceptron outputting real numbers.

TTR perceptron training

We can now formulate the perceptron training rule as updating a TTR record

ptrain(m, s, C) $=$ m but with
m.w$\leftarrow$ m.w$+\eta^n\cdot s$.sr$_C$
m.t$\leftarrow$ m.t$-\eta$

where

- m is a meaning (e.g. [right])
  - m.w:RealVector$_n$, m.t:Real
- s is a record type representing a take on a situation
- C is a sensor name, e.g. *pos*, corresponding to a perceptual category (e.g. position)
  - s.sr$_C$ is a sensor reading in s
- η^n is an n-dimensional real-valued vector where $\eta^n_m = \eta$ for all $m, 1 \leq m \leq n$, e.g. $\eta^2 = \begin{bmatrix}\eta & \eta\end{bmatrix}$

In the example above, for $\eta = 0.1$ we get

$[\text{right}]^{B\prime} = \text{ptrain}([\text{right}]^{B}, \text{s}_4^B, \textit{pos}) =$

$[\text{right}]^{B}$ but with

$$[\text{right}]^{B}.\text{w} \leftarrow \begin{bmatrix}0.800 & 0.010\end{bmatrix} + \begin{bmatrix}0.1 & 0.1\end{bmatrix} \cdot \begin{bmatrix}0.100 & 0.200\end{bmatrix}$$
$$[\text{right}]^{B}.\text{t} \leftarrow 0.090 - 0,1$$

which yields

$[\text{right}]^{B\prime}.\text{w} = \begin{bmatrix}0.810 & 0.030\end{bmatrix}$.
$[\text{right}]^{B\prime}.\text{t} = -0.010$.

B has thus updated the meaning of “right” by modifying the weight vector used by a classifier perceptron, based on the output of applying the dynamic semantics of “right” to B’s take on the situation.

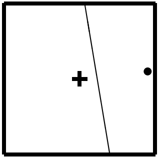
Agent B's initial take on the meaning of "right" , $[\text{right}]^B =$

$$\left[\begin{array}{l}
\text{w} = \begin{bmatrix}0.800 & 0.010\end{bmatrix} \\
\text{t} = 0.090 \\
\text{bg} = \left[\begin{array}{lcl} \text{sr}_{pos} & : & \text{RealVector} \\ \text{foc-obj} & : & \text{Ind} \\ \text{spkr} & : & \text{Ind} \end{array}\right] \\
\text{fg} = \left[\begin{array}{l}
\text{c}^{perc}_{right} = \left[\begin{array}{l}\text{sr}_{pos} = \text{bg.sr}_{pos} \\ \text{foc-obj} = \text{bg.foc-obj}\end{array}\right] : \\
\qquad \begin{cases} \text{right(bg.foc-obj)} & \text{if bg.sr}_{pos}\cdot\text{w} > \text{t} \\ \neg\text{right(bg.foc-obj)} & \text{otherwise} \end{cases} \\
\text{c}^{tell}_{right} = \left[\begin{array}{l} \text{str} = \text{"right"} \\ \text{spkr} = \text{bg.spkr} \\ \text{foc-obj} = \text{bg.foc-obj} \end{array}\right] : \text{right(bg.foc-obj)}
\end{array}\right] \\
\text{sfun} = \lambda x : \text{bg}(\text{fg}[\text{bg}/x]) \\
\text{dfun} = \lambda x \sqsubseteq \text{bg}(x \dot{\wedge} \text{fg}[\text{bg}/x])
\end{array}\right]$$

Agent B's revised on the meaning of "right", $[\text{right}]^{B'} =$

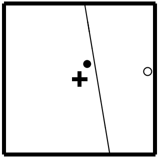
$$
\left[\begin{array}{l}
\text{w} = \begin{bmatrix} 0.810 & 0.030 \end{bmatrix} \\
\text{t} = \text{-0.010} \\
\text{bg} = \left[\begin{array}{lcl} \text{sr}_{pos} & : & \text{RealVector} \\ \text{foc-obj} & : & \text{Ind} \\ \text{spkr} & : & \text{Ind} \end{array}\right] \\
\text{fg} = \left[\begin{array}{lcl}
\text{c}^{perc}_{right} & = & \left[\begin{array}{l} \text{sr}_{pos} = \text{bg.sr}_{pos} \\ \text{foc-obj} = \text{bg.foc-obj} \end{array}\right] : \\
 & & \begin{cases} \text{right(bg.foc-obj)} & \text{if bg.sr}_{pos} \cdot \text{w} > \text{t} \\ \neg\text{right(bg.foc-obj)} & \text{otherwise} \end{cases} \\
\text{c}^{tell}_{right} & = & \left[\begin{array}{l} \text{str} = \text{“right”} \\ \text{spkr} = \text{bg.spkr} \\ \text{foc-obj} = \text{bg.foc-obj} \end{array}\right] : \text{right(bg.foc-obj)}
\end{array}\right] \\
\text{sfun} = \lambda x : \text{bg}(\text{fg}[\text{bg}/x]) \\
\text{dfun} = \lambda x \sqsubseteq \text{bg}(x \dot{\wedge} \text{fg}[\text{bg}/x])
\end{array}\right]
$$

A: “right”

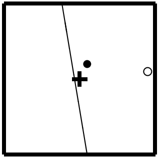


B: “okay”

A: “right”



B: “aha”



Outline

Conclusion

- The work presented here is part of a research agenda aiming towards a formal account of semantic coordination in dialogue.
- In this paper, we have presented a dynamic semantic approach to subsymbolic perceptual aspects of meaning.
- We have shown how a simple classifier of spatial information based on the Perceptron can be cast in TTR (Type Theory with Records).
- Furthermore, we have shown how subsymbolic aspects of meaning can be updated as a result of observing language use in interaction, thereby enabling fine-grained semantic plasticity and semantic coordination.

Future work

- There are many possible variants of the left-or-right game, which will be explored in future research.
- An obvious extension is to add more words (e.g. “upper” and “lower”) and some simple grammar (“upper left”, “lower right” etc) to explore compositionality of perceptual meanings.
- The left-or-right game can be extended by adding more interesting interaction patterns, including corrective feedback and explicit definitions.
- The capabilities of the agents could be extended by e.g. pointing.

Future work, con't

- Additional sensors and classifiers, e.g. for colour, shape and relative position, can be added.
- The fact that situations are stored as proofs can be useful in interactions where agent B rejects an utterance of by A and cites a previous situation when arguing for this rejection
  - "If this one here [pointing at object] was on the right, how can this one [pointing at other object] be on the left?"
- Explore how cases of type mismatch and background inconsistency can play out in (some more sophisticated version of) the left-or-right game.
- Add weights to all fields in records and record types, to account for further cases of gradience in linguistic representation

Brennan, S. E. and Clark, H. H. 1996.
Conceptual pacts and lexical choice in conversation.
Journal of Experimental Psychology: Learning, Memory and Cognition 22:482–493.

Clark, H. H. and Wilkes-Gibbs, D. 1986.
Refering as a collaborative process.
Cognition 22:1–39.

Clark, Eve V. and Wong, Andrew D. W. 2002.
Pragmatic directions about language use: Offers of words and relations.
Language in Society 31:181–212.

Clark, E. V. 2007.
Young children's uptake of new words in conversation.
Language in Society 36:157–82.

Coquand, Thierry; Pollack, Randy; and Takeyama, Makoto 2004.
A logical framework with dependently typed records.
Fundamenta Informaticae XX:1–22.

Garrod, Simon C. and Anderson, Anthony 1987.
Saying what you mean in dialogue: a study in conceptual and semantic co-ordination.
Cognition 27:181–218.

Healey, P.G.T. 1997.
Expertise or expertese?: The emergence of task-oriented sub-languages.
In Shafto, M.G. and Langley, P., editors 1997, *Proceedings of the 19th Annual Conference of the Cognitive Science Society*. 301–306.

Larsson, Staffan and Cooper, Robin 2009.
Towards a formal view of corrective feedback.
In Alishahi, Afra; Poibeau, Thierry; and Villavicencio, Aline, editors 2009, *Proceedings of the Workshop on Cognitive Aspects of Computational Language Acquisition, EACL*. 1–9.

Larsson, Staffan 2007.
Coordinating on ad-hoc semantic systems in dialogue.
In *Proceedings of the 10th workshop on the semantics and pragmatics of dialogue*.

Pickering, Martin J. and Garrod, Simon 2004.
Toward a mechanistic psychology of dialogue.
Behavioral and Brain Sciences 27(02):169–226.

Saxton, Matthew 1997.

The contrast theory of negative input.
Journal of Child Language 24:139–161.

Saxton, Matthew 2000.
Negative evidence and negative feedback: immediate effects on the grammaticality of child speech.
First Language 20(3):221–252.

Steels, Luc and Belpaeme, Tony 2005.
Coordinating perceptually grounded categories through language: A case study for colour.
Behavioral and Brain Sciences 28(4):469–89.
Target Paper, discussion 489-529.

Traum, David and Larsson, Staffan 2003.
The information state approach to dialogue management.

In Smith, Ronnie and Kuppevelt, Jan, editors 2003, *Current and New Directions in Discourse & Dialogue*. Kluwer Academic Publishers.