Formal semantics for perception

Staffan Larsson

Göteborgs Universitet
Box 200, SE405 30 Göteborg
sl@ling.gu.se

Abstract

A representation of subsymbolic perceptual aspects of meaning is proposed. We show how a simple classifier of spatial information based on the Perceptron can be cast in TTR (Type Theory with Records).

1. Introduction

In dynamic semantics, meanings are context-update functions which take an input context and return an updated (output) context. In this paper, a dynamic semantic approach to subsymbolic perceptual aspects of meaning is presented. We show how a simple classifier of spatial information based on the Perceptron can be cast in TTR (Type Theory with Records) (Cooper, 2012). A large variety of linguistic phenomena related to logical/symbolic meaning have already been addressed within this framework. Consequently, the TTR perceptron indicates that TTR may be a useful framework for integrating subsymbolic aspects of meaning in a way which allows us to keep around the accumulated insights from formal semantics.

2. The left-or-right game

As an illustration, we will be using a simple language game whose objective is to negotiate the meanings of the words "left" and "right". 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. The following procedure is repeated:

1. A places an object in the frame
2. B orients to the new object, assigns it a unique individual marker and orients to it as the current object in shared focus of attention
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"

3. Subsymbolic semantics

In this section, we will show how a TTR-based dynamic semantic account of meaning can be extended to incorporate subsymbolic aspects of meaning. Examples will be based on the left-or-right game introduced above.

3.1 Perceptual meanings as classifiers

We take the lexical meaning $[e]$ of an expression e to often contain not only compositional semantics but also perceptual meaning (at least for non-abstract expressions). By this we mean that aspect of the meaning of an expression which allows an agent to detect objects or situations referred to by the expression e. For example, knowing the perceptual meaning of "panda" allows an agent to correctly classify pandas in her environment as pandas. Likewise, 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 therefore think of perceptual meanings as classifiers of sensory input.

3.2 A TTR perceptron classifier

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 perceptron is a very simple neuron-like object with several inputs and one output. Each input is multiplied by a weight and if the summed inputs exceed a threshold, the perceptron yields 1 as output, otherwise 0 (or in some versions -1).

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

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

The basic perceptron returns a real-valued number (1.0 or 0.0) 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; for the moment we can think of this type as a "proposition".

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

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

Where fun will evaluate to

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

3.3 Situations and sensors

In the left-or-right game, we will assume that B's take on the situation includes readings from a position sensor (denoted "sr_{pos}") and 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.

Here is an example of B's take on the situation prior to playing a round of the left-or-right game:

$$\text{s}_1^B = \begin{bmatrix} \text{sr}_{pos}{=}\begin{bmatrix} 0.900 & 0.100 \end{bmatrix} & : & \text{RealVector} \\ \text{foc-obj=obj}_{45} & : & \text{Ind} \\ \text{spkr=A} & : & \text{Ind} \end{bmatrix}$$

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}.

3.4 Utterance interpretation

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

The meaning of a sentence is modelled as a function from a record (representing the context) of the type T_{bg} specified by the background meaning, to a record type representing the type of the foreground meaning, T_{fg}.

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

When updating an agent's take on the context, given a current take on the context T, if $T \sqsubseteq T_{bg}$ (i.e., T is a subtype of T_{bg}, which informally means that T minimally contains the information specified by T_{bg} but possibly also other information) then the updated context T' is $T \dot{\wedge} T_{fg}$ (but with any occurrences of bg in T_{fg} replaced by r). The $\dot{\wedge}$ is a *merge* operator such that $T_1 \dot{\wedge} T_2$ is T_1 extended with T_2.

$$\begin{bmatrix} \text{a=1:Int} \\ \text{b=2:Int} \end{bmatrix} \dot{\wedge} \begin{bmatrix} \text{c=3:Int} \end{bmatrix} = \begin{bmatrix} \text{a=1:Int} \\ \text{b=2:Int} \\ \text{c=3:Int} \end{bmatrix}$$

3.5 The meaning of "right"

We can now say what a meaning in B's lexicon might look like. 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" is represented thus:

$$[\text{right}]^B =$$
$$\begin{bmatrix} \text{w} = \begin{bmatrix} 0.800 & 0.010 \end{bmatrix} \\ \text{t} = 0.090 \\ \text{bg} = \begin{bmatrix} \text{sr}_{pos} & : & \text{RealVector} \\ \text{foc-obj} & : & \text{Ind} \\ \text{spkr} & : & \text{Ind} \end{bmatrix} \\ \text{fg} = \begin{bmatrix} \text{c}_{right}^{perc} = \begin{bmatrix} \text{sr}_{pos} = \text{bg.sr}_{pos} \\ \text{foc-obj} = \text{bg.foc-obj} \end{bmatrix} : \\ \begin{cases} \text{right(bg.foc-obj)} & \text{if bg.sr}_{pos} \cdot \text{w} > \text{t} \\ \neg\text{right(bg.foc-obj)} & \text{otherwise} \end{cases} \end{bmatrix} \end{bmatrix}$$

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 position sensor reading and a focused object foc-obj. In the fg field,the value of c_{right}^{perc} is a proof of either or right(foc-obj) or ¬right(foc-obj), depending on the output of the classifier perceptron which makes use of w and t. Here, right(y) is a perceptual "proposition" (a type constructed from a predicate), and objects of this type are proofs that y is (to the) right. As a proof of right(foc-obj) we count a "snapshot" of relevant parts of the situation, consisting of the current sensor reading and a specification of the currently focused object.

4. Contextual interpretation

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 s_1^B (see above).

To interpret A's utterance, after checking that $\text{s}_1^B \sqsubseteq [\text{right}]^B.\text{bg}$, B computes $[\text{right}]^B.\text{fg} \dot{\wedge}\ \text{s}_1^B$ to yield a new take on the situation s_2^B:

$$\text{s}_2^B = [\text{right}]^B \dot{\wedge}\ \text{s}_1^B =$$
$$\begin{bmatrix} \text{sr}_{pos}{=}\begin{bmatrix} 0.900 & 0.100 \end{bmatrix}\text{:RealVector} \\ \text{foc-obj=obj}_{45}\text{:Ind} \\ \text{spkr=A:Ind} \\ \text{c}_{right}^{perc} = \begin{bmatrix} \text{sr}_{pos} = \begin{bmatrix} 0.900 & 0.100 \end{bmatrix} \\ \text{foc-obj} = \text{obj}_{45} \end{bmatrix} \text{:right(obj}_{45}) \end{bmatrix}$$

Here, the classifier takes s_1^B to contain a proof of right(obj_{45}). For an account of learning in the framework proposed above, see (Larsson, 2011).

5. References

Robin Cooper. 2012. Type theory and semantics in flux. In Ruth Kempson, Nicholas Asher, and Tim Fernando, editors, *Handbook of the Philosophy of Science*, volume 14: Philosophy of Linguistics. Elsevier BV. General editors: Dov M. Gabbay, Paul Thagard and John Woods.

Staffan Larsson. 2011. The ttr perceptron: Dynamic perceptual meanings and semantic coordination. In *Proceedings of the 15th Workshop on the Semantics and Pragmatics of Dialogue (SemDial 2011)*, Los Angeles (USA).