A constraint-based model of grounded compositional semantics

Van den Broeck, Wouter J. M.
Sony Computer Science Laboratory, 6, rue Amyot,
Paris, 75005, France
wouter@csl.sony.fr

Abstract

This paper outlines a constraint-based system that enables artificial agents to interpret and conceptualise rich meaning which involves different concept types and semantic functions. Such compositional meaning consists of a network of semantic building blocks that bundle a semantic function together with the necessary concept grounding and learning methods. The semantic blocks are implemented as constraints, and the compositional meaning is represented as a constraint network. The interpretation of such meaning corresponds to constraint satisfaction. The conceptualisation is realised as a goal-directed construction of the constraint network. The concept acquisition is fully integrated in the interpretation and conceptualisation processes.

Introduction

A system that enables agents to talk about the world can be decomposed in three sub-systems: the *sensorimotor system*, the *conceptual-intentional system* and the *language system* (Hauser et al., 2002). Figure 1 shows how these sub-systems interact.

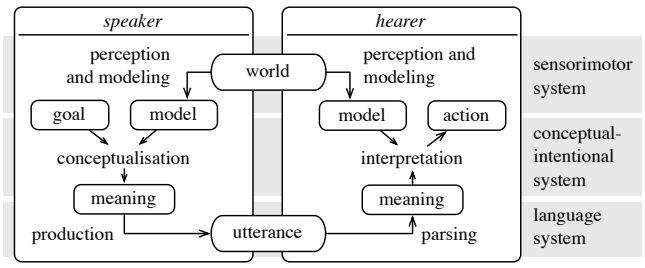

Figure 1: Overview of the interaction between the three sub-systems required for enabling artificial agents to communicate about the world through language.

The *sensorimotor system* takes care of the perception of the world and the construction of a model of that world. This world-model is used by the other sub-systems. The *language system* deals with the production of the utterance given the intended meaning, and the parsing of an utterance which yields the understood meaning. The third sub-system, the *conceptual-intentional system*, henceforth CIS, sits in between the language system and the sensorimotor system. It has to deal with the *representation*, *interpretation* and *conceptualisation* of meaning.

The *conceptualisation* process takes a speech-act goal and the world-model provided by the sensorimotor system. It composes the meaning that should be expressed in the utterance to be produced by the language system. The *interpretation* process takes the meaning reconstructed by the language system and interprets it in the context of the world-model provided by the sensorimotor system. The resulting action can range from executing the speaker's directive, or storing in memory the proposition in the speaker's assertive.

In this paper we focus on the conceptual-intentional system, and in particular on the question how such a system can be implemented for use in experiments involving language games (Steels, 1995).

To start we will consider the nature of the compositional meaning we want this system to be able to deal with. Such meaning consists of a network of semantic building blocks that take concepts as arguments. We will first look at the concepts and proceed with the semantic blocks.

Concepts

We will focus on speech-act goals that are concerned with the discrimination and/or description of objects in an observed scene. Figure 2 depicts a simple example scene which involves a number of objects with different shapes and sizes. If the speaker wants to draw the hearer's attention to object $o1$, then he/she could do so by saying "the pyramid". If the topic is rather object $o4$ then it could say "the big ball", while "the ball next to the big box" would do for object $o6$.

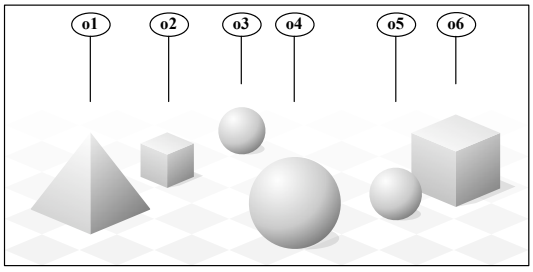

Figure 2: Simple example scene involving a number of objects, which are labeled for the purpose of the discussion.

Words like *ball*, *box*, *big*, *rightmost* and *next-to* each name a concept. Categories, prototypes, relations, events, roles, etc. are all different types of concepts.

Concepts can be used to discriminate specific objects by filtering them from a given context. A concept such as the shape prototype BALL can for instance be used to filter the objects that are ball-like, while a concept such as the comparison BIG can be used to filter the objects with a size larger than the average size.

Concepts that by themselves do not discriminate a topic can be combined. The phrase "the big ball" for instance, properly discriminates object $o4$ in the above example scene, even though there is more than one ball and several big things. There is however only one object that is both big and ball-like.

Concept grounding

The concepts need to be grounded in the sensorimotor functionality which interacts with the world. Different methods can be used for the grounding of concepts, for example neural networks are used in (Plunkett et al., 1992), probability density estimation in (Roy and Pentland, 2002), radial basis function networks in (Steels and Belpaeme, 2005), nearest neighbor (Belpaeme and Bleys, 2005), discrimination trees (Steels, 1996), event feature detectors (Siskind, 2001), etc.

A grounding method is minimally capable of assessing if some entity in the world-model belongs to some category. Each category for a particular grounding method corresponds to some particular set of *concept parameters*. Figure 3 lists some basic grounding methods and the kinds of parameters associated with the concepts grounded by the respective techniques.

Since no single grounding method is well suited for all types of concepts, the system needs to accommodate different grounding techniques.

Semantic functions

Concepts serve as arguments for *semantic functions* such as the context filtering discussed before. Other examples of semantic functions are: quantification as in "*the* ball" or "*some* boxes", set operations as in "the balls *and* the boxes" or "all balls *except* the rightmost", predication as in "the ball *is big*", negation as in "the box is *not* round", deictic reference as in "... *that* is round", etc. Note that different semantic functions can use the same concepts.

The artificial agents need to be able to autonomously interpret meaning that involves such semantic functions. Each semantic function thus requires a procedural implementation. This implementation takes the relevant concepts as arguments and calls the relevant grounding methods where needed.

Consider for example a semantic function that filters a set of entities according to a concept type that is grounded by means of a multi-layered perceptron. The application of this filtering involves the categorisation of each entity in the context by means of the perceptron, which is configured with the parameters – the weights – associated with the given concept. The results of these categorisations are then used by the semantic function to derive the filtered target-set[1].

Concept acquisition

Each agent has its own collection of concepts. These repertoires are furthermore not fixed. Agents need to be able to invent or learn new concepts or adapt existing ones. The nature of the learning methods depends on the type of the concerned concepts. Figure 3 lists a number of concept grounding methods and potential learning methods. Back propagation can for example be used with the multi-layered perceptron based grounding method.

grounding method	concept parameters	learning methods
k-NN	points, k	new point or shift points
ML perceptron	weights, θ	back-propagation
discrimination tree	segment	segmentation
...	...	...

Figure 3: This table lists some basic grounding methods and the corresponding concept parameters and learning methods. For the k-nearest-neighbours (k-NN) method, the parameters are one or more prototypical points in the data-space, and the (optional) value k. The learning method is either simply adding the positive example point or shifting the points based on positive and negative examples. For the multi-layered (ML) perceptron method, the parameters are the weights and (optionally) the threshold function, while the learning method is back-propagation. The third grounding method involves a discrimination tree (Steels et al., 2000), for which a concept corresponds to some segment (a node in the tree) or a set of segments.

A typical learning situation occurs when the speaker's utterance involves a word that the hearer does not know. Consider for instance the situation in which the speaker says "the frouple" to discriminate object $o1$ (the pyramid) in figure 2. The hearer does not know this word and indicates that it could not understand the utterance. The speaker could then draw the attention to the topic through other means, such as by pointing to it. This presents a learning opportunity for the hearer. It now knows the context and the topic, and could try to infer the concept that corresponds with the word "frouple". The candidate concepts are those that properly discriminate that topic. All candidates, or one chosen according to some heuristics such as the saliency, can then be passed to the learning method associated with the grounding method.

This inference of candidate concepts can be seen as a different operational mode of the involved semantic function. Where interpretation corresponds to taking a context

[1] or multiple candidate target-sets

source-set and a concept to produce a filtered target-set, here the semantic function takes a source-set and a filtered target-set, and infers the concepts that could account for the filtering of that target-set from the source-set.

Compositional meaning

Rich meaning involves different types of concepts and a variety of semantic functions that take these concepts as arguments. The conceptual-intentional system has to be able to both interpret and conceptualise such compositional meaning. The building blocks of these compositions each bundle a semantic function together with the necessary grounding and learning methods, as shown in figure 4. The involved functionality is wrapped in a uniform, abstract interface. This abstraction enables the semantic *composer* to transparently combine disparate underlying functionality.

The interface of a semantic block provides one slot for each argument that the involved semantic function needs to operate over. Compositional meaning is constructed by linking together the slots of multiple semantic blocks.

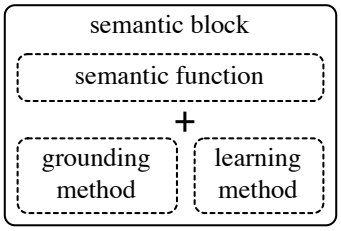

Figure 4: A semantic block combines a semantic function together with the necessary concept grounding and learning methods.

The tight coupling between the semantic functions and the grounding and learning methods affords a strong interaction between language use and concept formation. Such interaction is required to enable the structurally coupled evolution of language and concept repertoires.

Composition strategies

As mentioned earlier, the phrase "the big ball" properly discriminates object $o4$ in figure 2 because there is only one object that is both big and ball-like. The interpretation of this composition can be implemented by filtering in parallel the set of balls and the set of big things, and then taking the intersection of both sets.

This composition strategy is however not sufficient. Consider for instance the phrase "the big box" in the context of the scene shown below in figure 5. It discriminates object $o2$ even though the intersection of the set of big things $\{o1, o3\}$, and the set of box-like things $\{o2, o4\}$, is empty. Interpreting such phrase rather consists of first interpreting the noun relative to the context of the whole phrase. This yields a sub-context that consists of all boxes, i.e. $\{o2, o4\}$. Then the modifier is interpreted relative to this sub-context, which yields the bigger of both boxes, the intended topic.

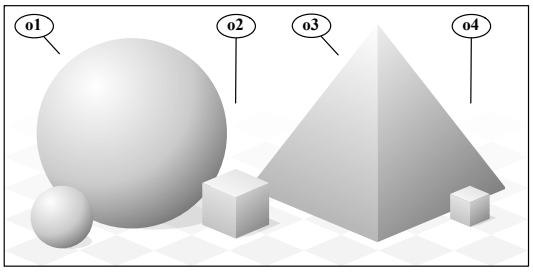

Figure 5: A scene with two big and three small objects.

The kind of context manipulation involved in the interpretation of a modifier-head structure can be made possible by providing (explicit) slots for the source-set (the input context) and the target-set (the filtered context) in the concerned semantic blocks. The modifier-head structure can then be attained by linking the target-set of the head's block to the source-set of the modifier's block. A block-diagram that represents this set-up is shown in figure 6.

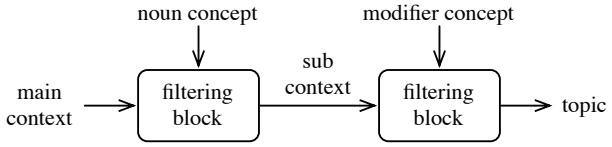

Figure 6: A semantic block-diagram for a noun + modifier phrase like "big box". The noun filtering block takes the main context and the noun concept and produces the filtered sub-context. This sub-context and the modifier concept are then used by the modifier building block to filter the topic.

Constraint networks

A semantic block generally supports several ways in which data flows in and out of that block. The concrete flow depends on the availability of values for the concerned slots. This availability is for instance different for a regular interpretation situation than for a learning situation. The ability to deal with different data-flows can be captured by implementing the semantic blocks as computational entities called *constraints*.

If the semantic blocks are implemented as constraints, then the compositional meanings correspond to constraint networks. Interpreting such meaning then corresponds to finding a solution for the constraint network, i.e. solving the *constraint satisfaction problem*.

A constraint can be represented as an n-slot predicate in which each slot is occupied by a variable. Multiple constraints form a network if slots from different constraints are occupied by the same variable.

Examples

Let's consider some examples of the interpretation and concept learning processes. These examples involve four types of semantic constraints, which are here represented

as n-slot predicates in which each slot is occupied by a variable. Multiple constraints form a network if slots from different constraints are occupied by the same variable.

The first two semantic constraints are *filter-set-prototype(target-set, source-set, prototype)* and *filter-set-size(target-set, source-set, comparison)*. They involve a filtering function such as described before. The first can filter the source-set by some prototype such as BALL or BOX. The second takes a comparison such as BIGGER-THAN, and retains in the target-set those objects from the source-set which are bigger than average.

The third block is *unique-element(object, set)*. This block asserts that the filler of the *set* slot is a set that contains one element; the filler of the *object* slot. It is used to cover the uniqueness of the topic. The fourth semantic block is *equal-to-context(set)*, which simply asserts that the filler of the *set* slot equals the set of objects in the observed context.

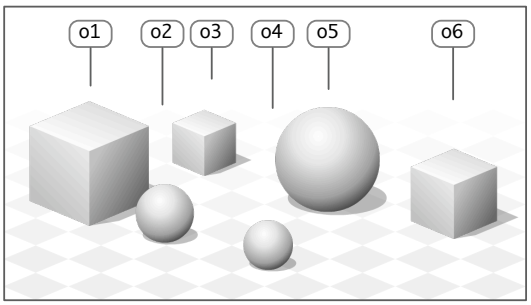

Figure 7: A scene with a number of object of varying size and shape.

Example 1: the big ball

Let's consider the semantic composition that discriminates object *o5* in the scene shown in figure 7. Combining these four constraints in a suitable composite meaning, gives the constraint network shown in figure 8.

```
{ equal-to-context(context),
  filter-set-prototype(set-1, context, prototype),
  filter-set-size(set-2, set-1, comparison),
  unique-element(topic, set-2) }
```

Figure 8: Semantic composition example 1

Interpretation Let's assume that the grammatical analysis of an utterance such as "the big ball" yields this composition plus the bindings: *prototype* ← BALL and *comparison* ← BIG, which are returned by the lexical look-up of "ball" and "big" respectively.

The semantic composition can now be interpreted by solving the constraint satisfaction problem. First the *equal-to-context* constraint binds the *context* variable to the complete set of objects in the scene, i.e. *context* ← $\{o1, o2, o3, o4, o5, o6\}$. Given the bindings for both the *context* and *prototype* variables, the *filter-set-prototype* constraint can infer a binding for *set-1*, i.e. the set of ball-like objects: $\{o2, o4, o5\}$. With this binding and the comparison, the *filter-set-average* constraint can now infer the binding *set-2* ← $\{o5\}$, since $o5$ is larger than the average size of the three balls. Finally, *unique-element* can correctly bind *topic* to $o5$, as such yielding the intended topic.

Acquisition Say we hear "the froople ball" but do not know the meaning of "froople". If we signal our misunderstanding to the speaker, and the speaker manages to draw our attention to the intended topic through other means, such as pointing, an opportunity for learning presents itself. We take the same semantic composition and fill in the known bindings: *prototype* ← BALL and *topic* ← $o5$. We can now again try to find a solution for the constraint network.

Applying the *unique-element* constraint gives the binding *set-2* ← $\{o5\}$. Applying the *equal-to-context* and *filter-set-prototype* constraints gives *set-1* ← $\{o2, o4, o5\}$. Given these bindings the *filter-set-size* block can try to abduct a comparison that could account for the filtering from the *set-1* to *set-2*. If this concept already exists in the inventory, a new entry between this concept and the form "froople" can be added in the lexicon. If it was not conceptualised before, it can also be added in the conceptual inventory.

Example 2: the ball next to the big box

As a second example we will assume the same context, but take $o2$ as the topic. We cannot easily find a semantic program that discriminates this topic using the same constraints as before. Let's therefore introduce an additional semantic block: *filter-set-relation(target-set, source-set, relation, referent)*. This block filters all elements from the source-set for which the relation does not apply with respect to the referent. The relations we consider here are spatial relations, such as NEXT-TO, or IN-FRONT-OF. This enables us to construct the semantic composition that corresponds to "the ball next to the big box", which properly discriminates the intended topic. The resulting composition is shown in figure 9.

```
{ equal-to-context(context),
  filter-set-prototype(set-1, context, proto-1),
  filter-set-size(set-2, set-1, comparison),
  unique-element(referent, set-2),
  filter-set-prototype(set-3, context, proto-2),
  filter-set-relation(set-4, set-3, relation, referent),
  unique-element(topic, set-4) }
```

Figure 9: Semantic composition example 2

For a regular interpretation the bindings are: *proto-1* ← BOX, *comparison* ← BIG, *proto-1* ← BALL, and *relation* ← NEXT-TO. Resolving the constraint network will first bind *referent* to $o1$ like in the previous example, and *set-3* to the set of balls, i.e. $\{o2, o4, o5\}$. Given these bindings

the *filter-set-relation* block can now select from *set-3* those elements which are 'next-to' the referent and bind this set, i.e. $\{o2\}$, to *set-4*, giving us the correct topic.

Goal-directed composition of constraint networks

The conceptualisation of a semantic composition corresponds to the construction of a constraint network. The input for this process is a communicative goal, e.g. 'discriminate topic *X* in the sensory context', and an inventory of primitive constraints. The resulting constraint network has to be coherent and fulfil the given goal when interpreted by the hearer. In order for the hearer to be able to properly interpret the decoded composition, all arguments that cannot be inferred should be expressed in the utterance. These *essential* arguments thus have to be representable in language, for instance as lexical forms.

Finding a suitable constraint network given some goal is a combinatorial problem. Blindly trying to link together various constraints in arbitrary configurations and checking if the results satisfy the requirements is not a viable strategy. We propose a structured, goal-directed strategy to manage the combinatorial explosion.

For a semantic composition to be useable, it must be resolvable given the essential arguments. All other bindings in the solution must be directly or indirectly inferable from this select set of bindings. In other words, there must exist a directed, non-cyclic dependency network among the bindings which reflects the inferential flow from the essential source bindings to the binding or bindings that represent or otherwise contribute to the communicative goal. The process of creating an appropriate semantic composition can be guided by this requirement.

Let's for example consider the construction of the semantic composition shown in figure 8. The initial goal is to discriminate object $o5$ from the sensory context shown in figure 7. We start the composition by introducing a variable and bind the topic to it. This binding is meant to be inferable during interpretation, so we need to add a constraint that can infer the binding. Most constraints however hold over more than one variable, which will need to be added. The bindings for these new variables also need to be either essential bindings or be inferable themselves. Introducing a new constraint to fulfil a goal might thus introduce new sub-goals, which need to be fulfilled recursively.

Let's say we add *unique-element(topic, set-2)* to infer the topic. This introduces a new sub-goal: find support for (the binding of) *set-2*. Adding *filter-set-average(set-2, set-1, comparison)* fulfils this sub-goal, but yields two new sub-goals: *set-1* and *comparison*. The comparison concept can be expressed in the utterance, but the set will have to be recursively dealt with.

A complete overview of the composition process is shown in figure 10. Each row represents a step in the process, starting with the initial step in the first row. The first column gives the goal for each step. The second column shows the 'action' taken to fulfil the goal, which is either a new constraint or an argument that has to be expressed in the utterance. The third column lists the sub-goals entailed by adding a constraint. Each of these sub-goals needs to be fulfilled in one of the subsequent rows.

goal	constraint or argument	subgoals
topic	unique-element(topic, set-2)	set-2
set-2	filter-set-average(set-2, set-1, comparison)	set-1, comparison
comparison	BIG	-
set-1	filter-set-prototype(set-1, context, prototype)	context, prototype
prototype	BALL	-
context	equal-to-context(context)	-

Figure 10: Goal directed composition

The composition process starts with the initial goal and ends when all the sub-goals that were introduced along the way, are fulfilled. For each goal there might be several constraints that could infer that goal. The composition shown in figure 10 thus represents but one particular path of potentially many. All these paths form a tree. Various strategies can be used to more efficiently explore this tree. We for instance apply an eager search strategy based on a heuristic that favours smaller compositions, with less unfulfilled goals and a smaller amount of essential arguments. We prune branches that involve a cyclic dependency and try to prune inconsistent branches as soon as possible by propagating the constraints where possible after each extension.

Finally we would like to note that this composition mechanism can also deal with situations in which the structure of the semantic composition was not fully understood. It can be used to hypothesise on a plausible completion of an incomplete network by adding constraints to account for bindings not yet accounted for in exactly the same way as outlined before.

Conclusion

A semantic building block bundles all cognitive functionality that concerns a particular concept type. This includes both the concept formation functionality and the semantic operations that for instance categorise a set of visual stimuli. By encapsulating the procedural details and providing a uniform, abstract interface, different concept grounding techniques can be transparently combined.

Semantic blocks establish an omni-directional relationship between a number of arguments, which can be naturally implemented as constraints. A semantic composition can then be represented as a constraint network. The declarative nature of such constraint networks permits a flexible control-flow. This affords a natural and uniform treatment of various compositional production, interpretation and learning needs, as was shown in the examples.

The grounding of both the concepts and the basic semantic operations is taken care of by the semantic blocks. The semantic compositions attain their grounding from

their components and the procedurally embodied constraint satisfaction framework.

In sum, the proposed model satisfies the requirements outlined in the introduction. A fully operational implementation of this model has been developed and can be demoed upon request.

Acknowledgements

The research presented in this paper builds on ideas first introduced in (Steels, 2000) and elaborated on in (Steels and Bleys, 2005).

This research is supported by Sony Computer Science Laboratory in Paris and the ECAGENTS project funded by the Future and Emerging Technologies programme (IST-FET) of the European Community under EU R&D contract IST-2003-1940. The information provided is the author's sole responsibility and doesn't reflect the Commission's opinions. The Commission is not responsible for any use that may be made of data appearing in this article.

References

Belpaeme, T. and Bleys, J. (2005). Explaining universal color categories through a constrained acquisition process. *Adaptive Behavior*, 13(4):293–310.

Hauser, M. D., Chomsky, N., and Fitch, W. T. (2002). The faculty of language: What is it, who has it, and how did it evolve? *Science*, 298:1569–1579.

Plunkett, K., Sinha, C., Moller, M. F., and Strandsby, O. (1992). Symbol grounding or the emergence of symbols? vocabulary growth in children and a connectionist net. *Connection Science*, 4:293—312.

Roy, D. K. and Pentland, A. (2002). Learning words from sights and sounds: a computational model. *Cognitive Science*, 26:113—146.

Siskind, J. M. (2001). Grounding the lexical semantics of verbs in visual perception using force dynamics and event logic. *Journal of Artificial Intelligence Research (JAIR)*, 15:31–90.

Steels, L. (1995). A self-organizing spatial vocabulary. *Artificial Life*, 2(3):319–332. (eds) Bedau, M.A. and Taylor, C.E. Cambridge, MA: The MIT Press.

Steels, L. (1996). Perceptually grounded meaning creation. In Tokoro, M., editor, *ICMAS96*. AAAI Press.

Steels, L. (2000). The emergence of grammar in communicating autonomous robotic agents. In Horn, W., editor, *ECAI2000*, pages 764–769, Amsterdam. IOS Press.

Steels, L. and Belpaeme, T. (2005). Coordinating perceptually grounded categories through language: A case study for colour. *Behavioral and Brain Sciences*, 28(4):469–89.

Steels, L. and Bleys, J. (2005). Planning what to say: Second order semantics for fluid construction grammars. In Bugarin Diz, A. and Reyes, J. S., editors, *Proceedings of CAEPIA '05. Lecture Notes in AI.*, Berlin. Springer Verlag.

Steels, L., Kaplan, F., McIntyre, A., and van Looveren, J. (2000). Crucial factors in the origins of word-meaning. In Dessalles, J.-L. and Ghadakpour, L., editors, *Proceedings of The 3rd Evolution of Language Conference*, pages 214–217, Paris. ENST 2000 S 002.