Aspects of descriptive, referential, and information structure in phrasal semantics

A construction-based model

Peter F. Dominey
Institut des Sciences Cognitives, CNRS UMR 5015

Phrasal semantics is concerned with how the meaning of a sentence is composed both from the meaning of the constituent words, and from extra meaning contained within the structural organization of the sentence itself. In this context, grammatical constructions correspond to form-meaning mappings that essentially capture this “extra” meaning and allow its representation. The current research examines how a computational model of language processing based on a construction grammar approach can account for aspects of descriptive, referential and information content of phrasal semantics.

Keywords: cognitive development, grammatical deixis, language acquisition, perceptual scene analysis, lexical categorization

1. Introduction

Part of the great expressive power of language is the ability to specify not only “who did what to whom,” but also the capability to nuance this thematic content with additional informational dimensions. Thus, the same event structure can be described in a number of different manners in order to emphasize different aspects of the contents. Consider, for example, the following state of affairs: accept(reviewer, paper), explain(paper, evolution), where meaning is encoded in a predicate(agent, patient) format. Depending on his or her discourse or pragmatic goals, the speaker can describe this situation with different sentences, including:

(1) The paper that explains evolution was accepted by the reviewer.
(2) The reviewer accepted the paper that explains evolution.
(3) The paper that was accepted by the reviewer explains evolution.
(4) Evolution was explained by the paper that was accepted by the reviewer.

From these examples, it can be observed that these different sentence structures direct or focus the attention of the listener on different aspects of the corresponding event semantics. In the context of “vocalize to localize,” this corresponds to a form of joint attention or grammatical deixis mechanism (Løvenbruck et al., in press) that is provided by phrasal semantics. A theory of language acquisition and processing should include a functional characterization of how phrasal semantics can be implemented and used to provide this expressive capability. The goal of the current research is to propose and validate a neuro-computationally plausible framework for how this could be implemented.

The problem of language acquisition can be posed in the following manner: given a set of <sentence, meaning> pairs, the child should learn which sentences are associated with which meanings, and should be able to generalize this knowledge to new sentences (e.g. Crain & Lillo-Martin, 1999: 56; Feldman et al., 1990). The child comes to this task equipped with some innate learning capabilities that are often referred to as the “initial state” or the language acquisition device (LAD). The <sentence, meaning> pairs are referred to as the primary linguistic data (PLD), and the result of learning is the adult state. One school of thought, associated with Chomsky (e.g., 1965, 1995) holds that the PLD is highly indeterminate and underspecifies the mapping to be learned. Thus, they propose that the LAD embodies a genetically pre-specified syntactic system or universal grammar (UG), and that language acquisition consists of setting the UG parameters to correspond to those for the target language. This implies the “continuity hypothesis,” which holds that via UG children have an adult-like syntactic system that they bring to the problem of language acquisition (Pinker, 1984, and see discussion in Tomasello, 2000). This school thus argues for a UG in which the essential structure of the grammar is innate, and they propose that what is learned are the values of parameters that identify the target language grammar within the UG framework. Partially because of this endowment of UG, this school advocates the characterization of grammars in terms of formal syntactic regularities that can be characterized largely independently of semantics and pragmatics.

(see Newmeyer, 1999, and papers from Tomasello, 1998) holds that the LAD does not contain a parameterized “universal grammar” but is rather a mechanism that learns the mapping between grammatical forms and meanings, (grammatical constructions) emphasizing the importance of communicative and social functions in language acquisition. This school places a much greater emphasis on the concrete relation between grammatical forms and meaning, and thus diminishes the independent significance of abstract generative syntactic rules.

In contrast with the continuity hypothesis, this framework is based in part on observations that the first grammatical constructions employed by infants appear more appropriately considered in terms of idiom-like linguistic gestalts that are initially fixed, and that through a progressive usage-based analysis become more open and productive (reviewed in Tomasello, 2000, 2003). In this context, the competence of the speaker is characterized as a structured inventory of grammatical constructions, rather than an abstract generative grammar. This view corresponds to the common ground for the construction grammar perspectives of Goldberg (1995, 1998, 2003), and Croft (2001) and the usage based approach to language acquisition of Tomasello (2003). Interestingly, Jackendoff (2002) considers that grammatical constructions can rightfully take their place as lexical items, thus blurring the distinction between lexical items and rules of the grammar.

Dominey (2000/2002) presented a construction based model of lexical and phrasal semantics that demonstrated capabilities for argument structure satisfaction, or thematic role assignment, with a relatively restricted set of active and passive grammatical constructions. The effort in that study was to examine the importance of the interaction between meaning and grammatical structure. In the current study, the exploration of the construction model is extended and demonstrated to account for multiple aspects of phrasal semantics.

Jackendoff (2003) has recently proposed a three tiered framework for describing phrasal semantics, or the manner in which sentence structure communicates meaning beyond that of the sum of the lexical elements. In this framework, phrasal semantics is organized into descriptive, referential and information/focus tiers. The descriptive tier includes thematic role assignment and associated argument satisfaction (i.e., the specification of “who did what to whom”), and the associated combinatorial ability to build relative clauses. In this context we will demonstrate how grammatical constructions indexed by word order and grammatical marking fulfill these functional criteria. The referential encompasses the existential or referential content of a sentence. In this
context we will demonstrate how the grammatical construction framework allows for resolution of particular types of pronoun reference and reflexive verb argument assignments. Finally, the information topic/focus tier includes representation of pragmatic focus that involves, for example, moving the thematic object to the head of the sentence in order to place a discourse focus on this element. Considering the following sentences: John pushed the block, and The block was pushed by John, one can see that these differ in their information content, due to the focus component. Again, by permitting this word ordering flexibility, phrasal semantics provides a powerful mechanism for grammatical deixis. In the next section the functional organization of the model is spelled out, and in Section 3 the performance of the model is described.

Part of the claim of the current research is that certain not-trivial aspects of the sentence-meaning relations in language can be learned. Such claims can be tested in the context of robotic systems that employ sensory perception for extracting meaning from the environment, which is then paired with verbal messages to provide input to the learning system. Section 3 will provide an overview of our debut in this line of research.

2. **Sentence to meaning mapping (SM²) model**

The model architecture is presented in Figure 1. From a behavioral perspective, during learning, the model is presented with <sentence, meaning> pairs, and it should learn the word meanings, and the set of grammatical constructions that define the sentence to meaning mappings in the input training set. During testing, the model should demonstrate that it can use this knowledge to understand new sentences that use the same lexicon, and the same set of grammatical constructions, but that were not presented in the training set. In particular the model should demonstrate systematicity, such that words that have only been experienced in particular syntactic roles (e.g., subject in an SVO sentence) will be correctly processed when they appear in new legal syntactic positions (e.g., the same word now as an object in an SVO sentence).

The functional organization of the model is based on the following principles: (1) Language acquisition can be characterized as learning the mappings from grammatical form to meaning (i.e., grammatical constructions) that should allow productive generalization with the learned constructions; (2) Within the sentence, the construction is encoded or identified by the relative configuration of open and closed class elements, that can thus be used as an
index by which the corresponding construction for that sentence type can be learned and retrieved. These concepts are presented in an overview in Figure 1. The following sections then describe the model in detail.

2.1 Input representations

The two inputs to the system are sentences, and meanings.

2.1.1 Sentence input

Sentences are encoded as linear sequences of words that are identified on input as being open or closed class elements. This lexical categorization is among
the early language-related perceptual distinctions learned by infants based on perceptible cues in the auditory signal (Shi et al., 1999, Höhle & Weissenborn, 2003, see papers in Morgan & Demuth 1996). We have recently demonstrated that in French and English, the temporal profile of the fundamental frequency ($F_0$) of the speech signal provides a reliable cue for categorizing open and closed class words (Blanc et al., 2003). Related simulation studies in which prosody is symbolically encoded have also demonstrated successful results (Morgan et al., 1996). The result of this early discrimination capacity applied to the child's target language is subsequently expressed in adulthood. Indeed, in adults, extensive data from event related potentials, brain imagery and psycholinguistic studies indicate that these lexical categories are processed by distinct and dissociated neurophysiological pathways (e.g., Kluender & Kutas 1993; Friederici, 1985; Pulvermüller, 1995; Brown et al., 1999).

In the model, words are represented as 25 element vectors, with content words coded with single bit in the range 1–16, and function words in 17–25. The content (or open class) words will be encoded in the Open Class Array (OCA) that contains 6 fields, each a 25-element vector with single bit encoding. The function (or closed class) words are encoded in a vector called the Construction Index described below. Additionally, each sentence is initiated by a closed class start symbol and terminated by a closed class end symbol.

2.1.2 Meaning input

If language acquisition is the learning of a mapping between sentences and meanings, then the infant must have some pre-linguistic capacity for representing this meaning. Well before their first birthday, infants can extract meaning from visual scenes and demonstrate the ability to understand physical properties of object interaction, and goal directed actions (e.g. Woodward, 1998; Carey & Xu, 2000; Bellagamba & Tomasello, 1999; Meltzoff, 1995; Mandler, 1996; Talmy, 1988; Kotovsky & Baillargeon, 1998). This implies the existence of conceptual representations of events that can be instantiated by non-linguistic (e.g., visual) perceptual input prior to the development of language. These conceptual representations will form the framework upon which the mapping between linguistic and conceptual structure can be built. This approach does not exclude the possibility that the conceptual representation capability will become more sophisticated in parallel with linguistic development (see Bowerman & Levinson, 2001 for a survey of the issue). It does require, however, that at least a primitive conceptualization capability that can deal with events in a predicate-argument format exists in a pre-linguistic state. Indeed, Fisher
(1996) has identified the requirement that event representations should take a predicate-argument structure that is related to the grammatical structure of the verb onto which they will be mapped. Likewise, in elaborating the structural relations between linguistic and conceptual forms Jackendoff considers that predicate/argument structure is a central feature of semantic structures (Jackendoff, 2002, p. 123). Similarly, this type of abstract predicate/argument event representation is central to the structure to meaning mapping for grammatical constructions as characterized by Goldberg (1995, 1998, 2003).

Thus, the “meaning” onto which the sentence is to be mapped takes this predicate(argument) form, encoded in the Scene Event Array (SEA) that consists of two sub-arrays that contain fields corresponding to action, agent, object, recipient/source. Each field is a 25-element vector with a single bit encoding. The SEA thus allows representation of the simple events (e.g., give, take, push, touch), as well as their combinations in hierarchical events, described below.

2.2 Learning word meanings: Lexical semantics

In the initial learning phases, the association between a word (in the OpenClassArray OCA) and its corresponding referent (in the SceneEventsArray SEA) is learned and stored in the associative memory of the WordToReferent matrix (Eqn 1). The parameter $\alpha$ specifies the influence of syntactic knowledge in "zooming in" on the appropriate word to referent mapping. In the initial configuration, prior to the accumulation of syntactic/grammatical construction knowledge, the term $\alpha$ is 1, and this learning simply associates every word with every element in the current scene. This exploits a form of cross-situational learning, in which the correct word-referent associations will emerge as those which remain constant across multiple sentence-scene situations (Siskind, 1996). In this manner the system can extract the cross-situational regularity that a given word will have a higher coincidence with the referent to which it refers than with other objects. This allows initial word learning to occur, which contributes to learning the mapping between sentence and scene structure (Eqn. 4, 5 & 6 below). Note that this first level has been addressed by Siskind (1996), Roy and Pentland (2000), and Steels (2001) and we treat it here in a relatively simple but effective manner, with more attention to the interaction between lexical and phrasal semantics.

Once this learning has occurred, knowledge of the grammatical structure, encoded in FormToMeaning can be used to “zoom in on” or identify the appropriate referent (in the SEA) for a given word (in the OCA). FormToMeaning is
an associative memory that specifies for each construction type, the mapping from elements in the OCA to elements in the SEA. Exploiting this knowledge allows the system to avoid mapping open class elements to the wrong scene referent elements. Functionally this corresponds to a zero value of \( \alpha \) in Eqn. 1. In this configuration, only the mapping between the word and its grammatically identified scene referent (i.e. the one specified in FormToMeaning) is strengthened. This corresponds to a form of "syntactic bootstrapping" in word learning. Thus, for the new word "gugle", knowledge of the appropriate grammatical construction for the sentence "John pushed the gugle" can be used to assign "gugle" to the object of push. LexLearningRate is a scalar valued parameter that specifies the “learning rate” or rate of change in weights in the WordToReferent matrix.

\[
\text{WordToReferent}(i,j) = \text{WordToReferent}(i,j) + OCA(k,i) * SEA(m,j) * \text{LexLearningRate} * \max(\alpha, \text{FormToMeaning}(m,k))
\]  

(1)

2.3 Mapping sentence to meaning: Phrasal semantics

The objective of phrasal semantics in the current context is to determine the mapping from sentence to meaning, particularly with respect to thematic role assignment and the related issues of phrasal semantics as outlined above. The learning task for the model is, given a set of \(<\text{sentence, meaning}>\) input pairs, to acquire the corresponding inventory of grammatical constructions that accounts for those sentences, and that can generalize to all new sentences within that set of grammatical constructions. In terms of the architecture in Figure 1 with the example sentence “The block was pushed by the triangle,” the underlying processes are defined in the following successive steps. First, words in the Open Class Array (block, pushed, triangle) are decoded into their corresponding scene referents (via the WordToReferent mapping described above) to yield the Predicted Referents Array (Eqn 2) that thus contains the translated words \(<\text{block, pushed, triangle}>\) while preserving their original order from the OCA.

\[
PRA(k,j) = \sum_{i=1}^{n} OCA(k,i) * \text{Word-to-World}(i,j)
\]  

(2)

The grammatical construction for the input sentence corresponds to a specific mapping between referents (in PRA) and the components of the meaning representation (in SceneEventArray SEA). This mapping is encoded in the Form-
ToMeaning array. The problem will be to store and retrieve, for each grammatical form, the appropriate corresponding FormToMeaning mapping, i.e. the construction. To solve this problem, we must extract from each grammatical form a unique corresponding Construction Index, based on lexical category, word order and grammatical marking (Bates et al., 1982). Then, the appropriate FormToMeaning mapping for each grammatical form can be indexed by its corresponding Construction Index. We first consider how the Construction-Index is generated, and then how it is associated with the appropriate FormToMeaning mapping.

The Construction Index (Eqn.3) encodes the grammatical structure of a sentence in terms of the function words and their relative position with respect to content words in the sentence. It is thus a re-coding of the sentence in which both position and identity of function words is preserved, while for content words, only position is preserved (yielding something like “___ was ___ by ___” for the example in Figure 1). Since each grammatical form or construction has a unique configuration of function and content words, with respect to their identity, order and relative position, the Construction Index will thus uniquely identify each distinct grammatical form. The Construction Index is a 25 element vector. Each function word is encoded as a single bit in the 25 element FunctionWord vector. When a function word is encountered during sentence processing, the current contents of Construction Index are shifted by n + m bits in a ring buffer (indicated by f_shift) where n corresponds to the bit that is on in the current FunctionWord vector, and m corresponds to the number of open class words that have been encountered since the previous function word (or the beginning of the sentence). Finally, a vector addition is performed on this result and the FunctionWord vector (i.e., bit n is set). In other words, for bit k in the current ConstructionIndex, the bit corresponding to \((k + n + m)\) modulo 25 will be set in the new ConstructionIndex, and finally bit n will be set. While this may seem obscure, the desired result is that the Construction-Index should uniquely code or represent distinct grammatical constructions as a function of their relative configurations of open and closed class words.

Equation 3 was designed to carry out this discrimination function purpose, and will be demonstrated to behave as desired, though clearly there may exist potentially superior alternatives to be explored in the future.

\[
\text{Construction Index} = f_{\text{shift}}(\text{Construction Index, (n + m)}) + \text{FunctionWord} \tag{3}
\]
The link between the Construction Index and the corresponding FormToMeaning mapping is established as follows. During training, as each input sentence is processed, we reconstruct the specific FormToMeaning mapping for that sentence (Eqn 4). The resulting FormToMeaningCurrent encodes the correspondence between word order that is preserved in the Predicted Referents Array PRA Eqn 2 (block, pushed, triangle in the example in Figure 1) and thematic roles in the SEA (pushed, triangle, block in the example in Figure 1). Note that the quality of FormToMeaningCurrent will depend on the quality of acquired word meanings in WordToReferent used to populate the PRA. Thus, syntactic learning requires a minimum baseline of semantic knowledge, corresponding to the “asyntactic first pass” discussed by Gillette et al. (1999).

Given the FormToMeaningCurrent mapping for the current sentence, we can now associate it with the corresponding Construction Index for that sentence (Eqn 5), storing this association in the ConstructionInventory associative memory.

\[
\text{FormToMeaningCurrent}(m,k) = \sum_{i=1}^{n} \text{PRA}(k,i) \times \text{SEA}(m,i) \quad (4)
\]

\[
\text{ConstructionInventory}(i,j) = (\text{ConstructionInventory}(i,j) + \text{Construction Index}(i) \times \text{Sentence-to-World-Current}(j) \times \text{PhrasalLearningRate}) / \text{Sum(ConstructionInventory)} \quad (5)
\]

Finally, once a construction has been learned via this mechanism, for new sentences we can extract the FormToMeaning mapping from the learned ConstructionInventory by using the Construction Index literally as an index into this associative memory, illustrated in Eqn. 6.

\[
\text{FormToMeaning}(i) = \sum_{i=1}^{n} \text{ConstructionInventory}(i,j) \times \text{ConstructionIndex}(j) \quad (6)
\]

It should also be noted that the associative memory in ConstructionInventory is subject to the standard hazards of simple associative memories of this type. In particular, for ConstructionIndex vectors that are similar, there may be retrieval errors. For this reason, we have also implemented ConstructionInventory in a more robust and functionally equivalent lookup table where ConstructionIndex acts as an index, and the appropriate FormToMeaning map is stored/retrieved. An advantage of this method is that constructions are
discretely coded and can be analysed post-hoc, e.g. the number of constructions required to account for an input corpus can be quantified.

In addition to simple <sentence, meaning> pairs such as <"The block pushed the ball", push(block, ball)>, we will also consider hierarchically structured pairs such as <"The block that pushed the ball touched the triangle", push(block, ball), touch(block, triangle)> that employs a relativised sentence and a dual-event scene.

To accommodate the dual scenes for such complex events, Eqns. 4–7 are instantiated twice each, to represent the two components of the dual scene. In the case of simple scenes, the second component of the dual scene representation is null. Likewise, there are two instances of the following data structures: ConstructionInventory, FormToMeaning, FormToMeaningCurrent, and SceneEventArray, to account for the dual event scenes, and the corresponding mapping mechanisms.

We evaluate performance by using the WordToReferent and FormToMeaning knowledge to construct for a given input sentence the “predicted scene”. That is, the model will construct an internal representation of the scene that should correspond to the input sentence. This is achieved by first converting the OpenClassArray into its corresponding scene items in the PredictedReferentsArray as specified in Eqn. 2. The referents are then re-ordered into the proper scene representation via application of the FormToMeaning transformation as described in Eqn. 7, after FormToMeaning is retrieved from the ConstructionInventory with the ConstructionIndex as described in Eqn. 6.

\[
PSA(m,i) = PRA(k,i) \times \text{FormToMeaning}(m,k)
\] (7)

When learning has proceeded correctly, the predicted scene array (PSA) contents should match those of the scene event array (SEA) that is directly derived from input to the model. We then quantify performance error in terms of the number of scene interpretation errors, or mismatches between PSA and SEA.

3. Model performance

We will now examine how this model of grammatical construction learning can address the issues of phrasal semantics as outlined in the introduction. Part of the limitation on the complexity of the grammar studied in Dominey (2000) was due to the simple structure of the meanings or scene representations that were employed, that consisted of a single event with three arguments. This
allows sentence types including active, dative, passive and dative passive. However, a sentence with a relativised subject, such as “The block that was pushed by the moon touched the triangle” corresponds in fact to two distinct events, that could be represented as push(moon, block), and touch(block, triangle). Here, we will first observe the benefits that a more complex scene representation can provide for the development of more complex grammatical structures, particularly relativised phrases.

3.1 Aspects of the descriptive and information tiers

As indicated above, the model should address aspects of the descriptive tier that includes thematic role assignment and associated argument satisfaction, as well as the combinatorial ability to build relative clauses. Likewise, it should address aspects of the information topic/focus tier that includes the capability for expression of pragmatic focus that involves, for example, moving the thematic object to the head of the sentence in order to place a discourse focus on this element.

Figure 1, and the description in Section 2.3 have described how the model performs simple argument structure satisfaction in the context of grammatical constructions. However, one of the hallmarks of human language is the ability to cope with hierarchical complexity in sentences. From a syntactic perspective, this complexity has been extensively analyzed, and the rules that govern the relations between components at different hierarchical levels are the object of an extensive body of research (e.g., Chomsky 1995). One can thus consider abstract hierarchical structure in the absence of meaning, as in autonomous syntax. In this absence of meaning, however, the purpose of hierarchical structure remains rather abstract. From the perspective of meaning, or semantics, however, the purpose of this hierarchical structure becomes strikingly functional. This is reflected in the language processing architecture suggested by Jackendoff (1999, 2002), in which the conceptual/semantic component has rich combinatorial hierarchical structure independent of (but mapped onto) syntactic structure.

Consider for example, the sequence of events depicted in the lower left corner of Figure 2. In these two scenes, the common element is “block.” Depending on the discourse focus (i.e. the head of the sentence) this complex scene can be described in different ways. We can consider the sentence where block is in the focus with an active verb yielding: “The block that was touched by the triangle pushed the circle.” Note that for the sake of consistency, we adopt the
Aspects of descriptive, referential, and information structure in phrasal semantics

convention that the referent of the relative clause is always the first of the two scene events. The same scene can also be described by a sentence that places the item “circle” in the discourse focus: “The circle was pushed by the block that was touched by the triangle.” Interestingly here we see that if focus is taken into account, then the apparent redundancy between the active and passive forms is eliminated.

An example of the processing of this relativised sentence “The block that was touched by the triangle pushed the circle” is provided in Figure 2. As for simple (single event) sentences, the OpenClassArray elements are translated to their referent via the lexical semantics information in WordToReferent, thus populating the PredictedReferentsArray. The meaning component of the <sentence, meaning> pair for this sentence corresponds to two distinct events: touch(triangle, block), push(block, circle). These two event representations are linked by the common element block, thus forming a hierarchically linked semantic representation that corresponds to the hierarchical structure of the relative sentence. The task now is to map these PredictedReferentsArray elements onto the dual event structure of the SceneEventsArrays. As mentioned above in the model description, to account for the dual event scenes and the corresponding mapping mechanisms there are two instances of the following data structures: ConstructionInventory, FormToMeaning, FormToMeaningCurrent, and SceneEventArray. Thus, the ConstructionIndex will retrieve FormToMeaning

Figure 2. Example of Relativized Sentence Processing. The hierarchical structure of the sentence is reflected in the structure of the semantic representation. See text.
mappings corresponding to the two scene events. Each will respectively map the contents of the PredictedReferentsArray onto the appropriate elements in the two SceneEventsArrays. Note once again that in the current example, the referent *block* maps onto different thematic roles in the two events — reflecting the linked hierarchical structure of the relative phrase.

The current study thus exposes the model to grammatical construction Types 1–26 of the Appendix. In this experiment ConstructionInventory functions as a lookup table, functionally equivalent to the associative memory but more efficient, where ConstructionIndex acts as an index, and the appropriate SentenceToWorld map is stored/retrieved. The model learns the 26 sentence types without errors, demonstrating that the ConstructionIndex is robust to the structural variability in these sentence types. In other words, each of the different grammatical constructions generates a distinct ConstructionIndex as defined by Eqn. 3, and thus the appropriate FormToMeaning mapping can be stored and retrieved using the ConstructionIndex as an index into the ConstructionInventory.

In this manner, the model has been demonstrated to generalize without error to new sentences that (1) use words that have been learned and stored in the WordToReferent associative memory, and that (2) use grammatical constructions that have been previously learned and stored in the FormToMeaning associative memory.

Returning to the examples of relativised sentences (1–4) presented in the introduction, we can now see in the Appendix that they correspond to construction types 8–11. In these elements of the appendix, we see the grammatical constructions defined in terms of the sentence structure “frames” and the corresponding semantic structure or meaning “frames”. This demonstrates that the concept of grammatical constructions as form to meaning mappings quite adequately captures the phrasal semantic requirements for the expression of relativised noun phrases, as well as the liberation from fixed word order (e.g. in passive forms) that allows a form of syntactic deixis.

3.2 Aspects of referential tier and beyond

From the perspective of the referential tier, the model should also demonstrate how the grammatical construction framework allows for resolution of pronoun reference and reflexive verb argument assignments. As illustrated above, “dual” events in the meaning representation allow the use of hierarchical meanings that correspond to relative clauses. The current experiment demonstrates how
dual events also support additional sentence types including: conjoined (John took the key and opened the door), reflexive (The boy said that the dog was chased by the cat), and reflexive pronoun (The block said that it pushed the cylinder) sentence types (Types 27–38).

The consideration of these sentence types compels us to address the question of how their meanings are represented. Conjoined sentences are represented by the two corresponding events, e.g., \textit{took(John, key), open(John, door)} for the example above. Reflexives are represented, for example, as \textit{said(boy), chased(cat, dog)}. This assumes for reflexive verbs (e.g., \textit{said}, \textit{saw}), that the meaning representation includes the second event as an argument to the first. Finally, for the reflexive pronoun types, in the meaning representation the pronoun’s referent is explicit, as in \textit{said(block), push(block, cylinder)} for “The block said that it pushed the cylinder.”

Note that the pronoun is treated as a closed class element and thus encoded in the ConstructionIndex, and not in the OpenClassArray. The net result is that the behavior of the system is correct — the sentence is reliably mapped onto its meaning. At a finer level, however, the referent for “it” is not explicitly represented. To begin to address this issue, one could channel incoming pronouns both to the ConstructionIndex, and directly to the predicted referents array without decoding of their lexical semantics. This would then potentially allow on-line binding between pronouns and their corresponding scene elements via the FormToMeaning mapping.

Based on <sentence, meaning> pairs as thus specified, the model learns the appropriate FormToMeaning mappings for the ensemble of sentence types in the Appendix. This demonstrates that the 38 sentence types are structurally distinct, as the system extracts unique ConstructionIndices for each of them. This allows, for each type, the binding of the appropriate FormToMeaning mapping to the corresponding ConstructionIndex. In the first exposure to a new construction type, the model matches the predicted referents with the scene event elements to determine the current form to meaning mapping. This mapping is stored in the ConstructionInventory, indexed by the ConstructionIndex unique to that sentence type. The observation here that the model can learn the 38 constructions from the appendix confirms that at least for these constructions there is a unique pattern of closed class items for each, and that this can be used to store and retrieve the form to meaning mappings.
3.3 Robot language acquisition in the construction framework

Given the demonstrated ability of the model to learn sentence to meaning mappings, we have recently started to explore the use of this model in a robot language learning context (Dominey, 2003; Dominey & Boucher, 2005). In these experiments, a human experimenter manipulates toy blocks in the field of view of a computer vision system, and simultaneously narrates her actions, something like “The block was pushed by the triangle that touched the moon”. This involves the automatic extraction of sentences from speech using standard human language technology tools, and the extraction of meaning from visual scenes. For meaning extraction, we use an approach similar to that of Siskind (2001) but much simpler, with off-the-shelf (SmartVision Panlab) color based computer vision for object recognition and tracking in order to extract contact events between objects in dynamic scenes. Then, events such as touch, push, take, give are parsed from the stream of contacts based on the definition of each of these event types in terms of a specific pattern of contact or contact sequence. Causality is attributed as a function of relative velocity of objects involved in a contact, i.e. the object that was moving faster towards the other object “did it”. The events are coded in predicate(argument) form as described above. Thus, from live human generated and narrated events, our robotic system can extract <sentence, meaning> pairs, and use the construction based model to learn the underlying grammatical constructions (Dominey, 2003a). This provides a concrete demonstration, as proposed by Goldberg (1995), of the tight correspondence between the structure of perceptual events that are basic to human experience, and the constructions for the corresponding basic sentence types.

4. Discussion

The sentence to meaning mapping model presented here embodies central aspects of construction grammar (Goldberg, 1995, 1998; Croft 2001) and its application in a usage-based characterization of language acquisition (Tomasello, 2003), along with central tenets of the cue competition and coalition framework (Bates & MacWhinney, 1982). This can be expressed as the following principles: (P1) Language acquisition can be characterized as learning constructions or mappings from grammatical form to meaning that allow productive generalization with the learned constructions; (P2) Within the sentence, the construction is encoded or identified by the relative order or configuration
of open and closed class elements that can thus be used as an index by which the corresponding construction for that sentence type can be learned and retrieved. Evaluation of the model with respect to its ability to fulfill the requirements of a lexical semantics system thus serves as a form of evaluation of these theories (to the extent that they are embodied in the model). The model has developed in a line of research that attempts to explain aspects of language processing in the context of cognitive sequence processing (Dominey et al., 2003; Dominey & Ramus, 2000; Lelekov et al., 2000a,b; Hoen et al., 2003). While the current exposition has been limited to English, we have recently demonstrated that the two principals of the model are applicable in a cross-linguistic validation to Japanese (Dominey & Inui, 2004).

The stated objective of this research was to examine the abilities of a construction-based model of language processing to accommodate a well-defined subset of the functional requirements of phrasal semantics. Clearly the whole job of phrasal semantics is an immense research project, hence the liberal use of the word “aspects” to signify the limited nature of the current analysis. Given this proviso, let us consider to what extent the objectives have been realized. With respect to the descriptive tier, we have seen that the construction model handles a variety of abstract constructions, employing a novel and effective method for argument structure satisfaction, or thematic role assignment. In this context, the system also demonstrates a novel and effective method for accommodating relative clauses in NPs, assigning an important role to the hierarchical structure of meaning in driving that of grammatical structure. This enters into the current discussion concerning the hierarchical and recursive structure of different dimensions of linguistic structure. In this context, Hauser et al. (2002) have proposed that syntax alone possesses a capability for recursive structure, while Jackendoff (1999, 2003) argues that phonology, semantics and syntax are all independently recursive. The current study suggests that hierarchical structure in semantic or conceptual representations provides the structural framework onto which sentence structure is mapped. This could be extended to propose that the recursive and compositional structure of syntax is derived from that of the conceptual structure that it expresses.

With respect to the information/focus tier, the model clearly demonstrates that the adopted grammatical construction approach is quite adequate for allowing the use of multiple non-canonical word orderings in order to rather precisely manipulate the focus or informational content of different grammatical constructions. Similarly, in the context of the referential tier, the system demonstrates the capability for reflexive and reflexive pronoun constructions.
From a developmental perspective, the construction paradigm provides an easy access for the infant into the world of utterance level language. In the earliest stages of utterance understanding, the child appears to treat sentences as idiom-like holophrases, gradually liberating these fixed constructions with increasing abstraction or schematicity. The resulting abstract construction capability provides the advantage of an ability to easily acquire a variety of construction types that allow systematic generalization to new sentences within the domain of the learned constructions. This is achieved with a striking minimum of “syntactic” machinery, that is replaced by structural mapping capabilities. As it stands the system suffers one significant limitation. The limitation is that all new constructions must be acquired by learning. That is, the system must be exposed to one <sentence, meaning> pair that is representative of the new construction in order to acquire the mapping. This is likely the case for children up to around two years of age (Tomasello, 2000, 2003; Clark, 2003). But clearly, the human language capacity includes the ability to produce and to comprehend sentences derived from novel grammatical constructions with no previous exposure to those constructions. Addressing this problem will require “fractionating” the current level of treatment of grammatical constructions into smaller units that would include noun phrases (Miikkulainen, 1996). This would allow pattern finding mechanisms to operate at the level of these sub-phrasal construction components that could then be recombined to provide an on line construction generation capability. In the mean time, the current research advances the current state of affairs by demonstrating that a model of language processing based on tenets from construction grammar (Goldberg, 1995, 1998) and the coding of phrasal semantics (Bates & MacWhinney, 1982) can begin to account for interesting aspects of phrasal semantics (Jackendoff, 2003).

Acknowledgements

This work was supported by the ACI for Integrative and Computational Neuroscience, the European Eurocores Origin of Man, Language and Languages Project, and the HFSP.

References


Bowerman, M. (1996). Learning how to structure space for language: A crosslinguistic per-


Carey S., & Xu F. (2001). Infant’s knowledge of objects: Beyond object files and object track-


tion*. Malden, MA: Blackwell.


Dominey, P.F. (2003). Learning grammatical constructions in a miniature language from

Dominey, P.F., & Boucher, (2005). Developmental stages of perception and language acqui-


Dominey, P.F., Inui, T. (2004). A developmental model of syntax acquisition in the Con-
struction Grammar framework with cross-linguistic validation in English and Japanese.

Dominey, P.F., & Lelekov, T. (2000). Nonlinguistic transformation processing in agrammati-


Haauser, M., Chomsky, N. & Fitch, T. (2002). The faculty of language: What is it, who has it, and how did it evolve? Science, 298(5598), 1569–79


**Author's address**

Peter F. Dominey  
Institut des Sciences Cognitives  
CNRS UMR 5015  
67, Boulevard Pinel  
F-69675 BRON Cedex  
France  
Email: dominey@isc.cnrs.fr

**About the author**

Peter Ford Dominey earned the BA at Cornell University in Cognitive Psychology and Artificial Intelligence in 1984. In 1989 and 1993 respectively he obtained the MSc and PhD in Computational Neuroscience from the University of Southern California, Los Angeles. From 1984 to 1986 he was a Software Engineer at The Data General Corporation and from 1986 to 1993 he was a Systems Engineer at the Jet Propulsion Laboratory in Pasadena, CA. From 1993 to 1997 he was a post-doctoral fellow at INSERM U94 in Lyon France, and in 1997 he became a tenured researcher in the CNRS. His research addresses understanding and simulating the neurophysiology of cognitive sequence processing and language, and their application to robot cognition.

**Appendix: Sentence type data base**

Sentences in the experiments used the nouns “block, moon, cylinder, dog, cat, ball” and verbs “touch, push, take, give, said” Closed class words included “that, to, from, by, was, it, itself, and ”. The different grammatical structures represent a subset selection of the possible constructions for simple and dual sentences in English. The model does not currently account in a generalized way for verb tense morphology. The current sentences were all in the past tense.

**Simple Event Sentences**

1. Agent action object. (Active) E.g. John pushed the block.  
   Action(agent, object) e.g. push(John, block).
2. Object was actioned by agent. (Passive) E.g. The block was pushed by John.  
   Action(agent, object), e.g., push(John, block).
3. Agent actioned object to recipient. (Dative) E.g., John gave the block to Bill.  
   Action(agent, object, recipient), e.g., gave(John, block, Bill)
4. Object was actioned to recipient by agent. (Dative passive)
E.g. The block was pushed to Bill by John.
Action(agent, object, recipient), e.g., pushed (John, block, Bill).

5. Agent action recipient object. E.g., John gave Bill the block.
Action(agent, object, recipient), e.g., gave(John, block, Bill).

**Double event relatives**

6. Agent1 that action1ed object2 action2ed object3. (Relative agent).
Action1(agent1,object2), Action2(agent1,object3)

7. Object3 was action2ed by agent1 that action1ed object2.
Action1(agent1,object2), Action2(agent1,object3)

8. Agent1 that action1ed object2 was action2ed by agent3
Action1(agent1,object2), Action2(agent3,object1)

9. Agent3 action2ed object1 that action1ed object2
Action1(agent1,object2), Action2(agent3,object1)

10. Object2 that was action1ed by agent1 action2ed object3
Action1(agent1,object2), Action2(agent1,object3)

11. Object3 was action2ed by agent2 that was action1ed by agent1
Action1(agent1,object2), Action2(agent2,object3)

12. Object2 that was action1ed by agent1 was action2ed by agent3
Action1(agent1,object2), Action2(agent3,object2)

13. Agent3 action2ed object2 that was action1ed by agent1
Action1(agent1,object2), Action2(agent3,object2)

14. Object3 was action2ed to recipient4 by agent1 that action1ed object2
Action1(agent1,object2), Action2(agent1,object3,recipient4)

15. Agent1 that action1ed object2 was action2ed to recipient4 by agent3
Action1(agent1,object2), Action2(agent3,object1,recipient4)

16. Agent3 action32ed object4 to recipient1 that action21ed object2
Action1(agent1,object2), Action2(agent3,object4,recipient1)

17. Object4 was action32ed from agent3 to recipient1 that action21ed object2
Action1(agent1,object2), Action2(agent3,object4,recipient1)

18. Object2 that was action1ed by agent1 action2ed object3 to recipient4
Action1(agent1,object2), Action2(agent1,object3,recipient4)

19. Agent3 action2ed object4 to recipient2 that was action1ed by agent1
Action1(agent1,object2), Action2(agent3,object4,recipient2)

20. Agent1 that action1ed object2 to recipient3 action2ed object4
Action1(agent1,object2,recipient3), Action2(agent1,object4)

21. Object4 was action2ed by agent1 that action1ed object2 to recipient3
Action1(agent1,object2,recipient3), Action2(agent1,object4)

22. Agent4 action2ed object1 that action1ed object2 to recipient3
Action1(agent1,object2,recipient3), Action2(agent4,object1)

23. Object1 that action1ed object2 to recipient3 was action2ed by agent4
Action1(agent1,object2,recipient3), Action2(agent4,object1)

24. Agent2 that was action1ed by agent1 to recipient3 action2ed object4
Action1(agent1,object2,recipient3), Action2(agent1,object4)

25. Agent4 action2ed object2 that was action1ed by agent1 to recipient3
Action1(agent1,object2,recipient3), Action2(agent4,object2)
26. Agent1 that acted object2 acted object3 to recipient4
Action1(agent1, object2), Action2(agent1, object3, recipient4)

**Dual event Conjoined**

27. Agent1 action1 object1 and object2. (Active conjoined object)
Action1(agent1, object1), Action1(agent1, object2)
28. Agent1 and agent3 action1ed object2. (Active conjoined agent)
Action1(agent1, object2), Action1(agent3, object2)
29. Agent1 action1ed object2 and action2 object3. (Conjoined)
Action1(agent1, object2), Action2(agent1, object3)

**Dual event Reflexive**

(Note that action1r corresponds to reflexive action such as “see” or “think”.)
30. Agent1 action1r that agent2 action2ed object3. (Simple reflexive)
Action1r(agent1), Action2(agent2, object3).
31. Agent1 action1ed itself. (Simple active reflexive)
Action1(agent1, agent1).
32. Agent1 action1r that agent2 action2ed itself. (Reflexive simple noun phrase).
Action1r(agent1), Action2(agent2, agent2).
33. Agent1 action1r that agent2 action2ed it. (Pronoun simple noun phrase).
Action1r(agent1), Action2(agent2, agent1).
34. Agent1 action1r that it action1ed object2.
Action1r(agent1), Action2(agent1, object2).
35. Agent1 action1r that object3 was action2ed by agent2.
Action1r(agent1), Action2(agent2, object3).
36. Agent1 action1r that agent2 action2ed object3 to recipient4.
Action1r(agent1), Action2(agent2, object3, recipient4).
37. Agent1 action1r agent2 action2ed object3 to recipient4.
Action1r(agent1), Action2(agent2, object3, recipient4).
38. Object2 object3 were action1ed to recipient4 by agent1.
Action1(agent1, object2, recipient4), Action1(agent1, object3, recipient4)