Ontology driven contextual best fit in Embodied Construction Grammar

Jesús Oliva[a], Jerome Feldman[b], Luca Gilardi[b], Ellen Dodge[b]

[a]*Bioengineering Group. Consejo Superior de Investigaciones Científicas - CSIC. Carretera de Campo Real, km. 0,200. La Poveda, Arganda del Rey. CP: 28500. Madrid, Spain.*
jesus.oliva@csic.es
[b]*International Computer Science Institute. 1947 Center Street Suite 600, Berkeley, CA 94704*

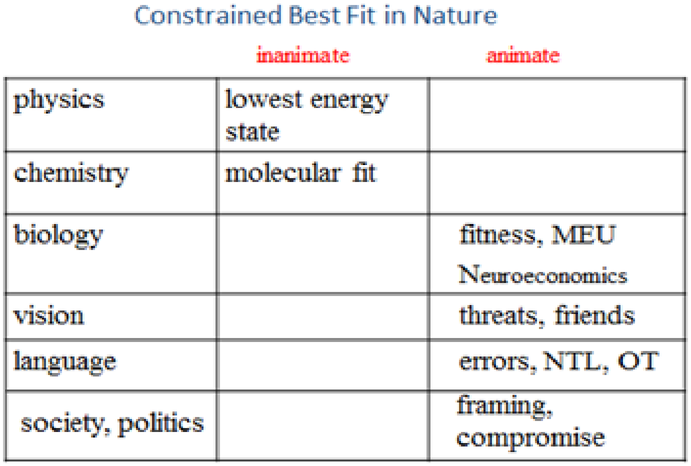
Abstract. Constraint satisfaction has been central to the ICSI/UC Berkeley Neural Theory of Language (NTL) project, but this aspect has not previously been emphasized. The ECG Analysis program combines constraints from several aspects of the formalism: deep semantic schemas, embodied constructions and ontological knowledge. In this chapter we focus on some applications of deep semantic constraints that extend the Embodied Construction Grammar formalism (ECG) and Analyzer. The first example is a shallow reference resolution method that is based on the combination of the recency principle with syntactic and semantic compatibility between the anaphor and the referent. The method has been implemented and tested as part of a system capable of understanding Solitaire card-game instructions, with promising results. Similar deep ontology-driven constraint satisfaction techniques can be exploited to handle many cases of Noun-Noun compounds and metaphorical constructions. Implemented examples of these are also presented.

1. Introduction

From a sufficiently general perspective, Constraint Satisfaction (CS) can be seen as one of the most fundamental processes in nature. A compact version of this story is depicted in Figure 1. This employs a *general* notion of CS, not any specific mechanism such as constraint *solution* in logic programming. In Figure 1, the abbreviation MEU, stands for "maximizing expected utility" a concept that is central to evolution and animal behavior. The term OT refers to Optimality Theory (Prince and Smolensky 2004), which uses best-fit CS techniques in a very different theory than employed in the NTL work described here.

Most relevant here is the fact that language understanding, like all perception, involves constrained best-fit of the input to the context and goals of the perceiver. This has been understood for some time (Feldman 2006) and plays a central role in the analysis module of the ECG system for semantically driven natural language understanding, shown in Figure 2. The language input to the system is analyzed using the best-fit analyzer to produce a semantic representation called the *SemSpec* (see details below). Then, the Specializer tries to extract the task-relevant meaning from that structure and passes this information as N-tuples to the Application side. One instance of this architecture is the system for understanding language about card games, described in Section 5.

One powerful example of best-fit CS in language understanding arises in languages, such as Mandarin, where almost any word can be omitted from an utterance if it is available from context. Partially to handle such situations, Bryant (2008) built a Bayesian best-fit Discourse Analyzer (middle left of Figure 2) that can determine the best semantic analysis, even for quite sparse input, like the Mandarin equivalent of "give Auntie". The constrained best-fit process combines three posterior probability scores. The first is close to a conventional stochastic context free grammar. The second score is an estimate of the (deep) semantic compatibility of fillers for various constructional roles and the third score estimates the goodness of fit for contextual elements not explicitly mentioned.



Constrained Best Fit in Nature

	inanimate	animate
physics	lowest energy state	
chemistry	molecular fit	
biology		fitness, MEU Neuroeconomics
vision		threats, friends
language		errors, NTL, OT
society, politics		framing, compromise

Figure 1. The Ubiquity of Constrained Best Fit

More generally, language understanding is highly context dependent. In particular, anaphors are constantly used in order to avoid unnecessary repetitions of particular words or structures. The meaning of many elements of each sentence and, by extension, the meaning of each sentence, depends on the meaning of previous utterances. Examples of this are pronouns (like *he*) or definite noun phrases (like *the boy*). The reference resolution task consists of linking these semantically undefined structures (the anaphor) to an entity previously found in the discourse (the antecedent) to which they refer. Therefore, reference resolution methods constitute a very important part of any language understanding system. This importance has attracted a great deal of research starting from the beginnings of the field of natural language processing, but perfect performance is still out of reach.

Many approaches have been tried for reference resolution[1]. Knowledge-based systems (from the first reference resolution methods (Hobbs, 1976, 1978) to some recent approaches (Asher and Lascarides, 2003)), were the first approaches but not the only ones since they are very complex systems, difficult to build, and they lacked robustness. Heuristic systems (Lappin and Leass, 1994; Mitkov, 1998) tried to solve those problems using designed heuristics to avoid the complexity of previous systems. Finally, machine learning systems reformulated the reference resolution task as a binary classification problem. An early approach of this kind was presented by Soon

[1] See (Kehler 2002) for an introduction

et al. (2001) and was followed by many other researchers introducing variations on that former algorithm (Ponzetto and Strube, 2006; Ng and Cardie, 2002; Ng, 2007).

Despite this large amount of work, naïve reference resolution methods (Hobbs, 1978; Mitkov, 1998) still have very good performance. These methods are based on selecting the most recent antecedent that is grammatically compatible with the anaphor. Our method is also based on combining anaphor referent compatibility with the recency principle (humans tend to select antecedent candidates based on their recency in discourse (Lappin and Leass, 1994)). However, we use a deeper kind of compatibility not restricted to grammatical features. We introduce some semantic features (such as the functionalities of the concepts referred to by the anaphor and the antecedent) in order to improve the performance of the method. This chapter focuses on structural and conceptual issues; no large scale performance studies have been done and none will be on these isolated tasks.

All this work is done using the framework of Embodied Construction Grammar (ECG) (Feldman, 2006). ECG is a formalism for representing linguistic knowledge in the form of construction-based grammars. This formalism allows us to transfer much of the work load of the reference resolution method to the design of the grammar and the ontology. The use of those structures extends the probabilistic best-fit analyzer implemented for ECG (Bryant, 2008; Bryant and Gilardi, 2013). The formalism and its implementation are central to this chapter.

In particular, the reference resolution method presented in this paper has been developed as a part of an ongoing effort of the Neural Theory of Language (NTL) project with the objective of implementing a system that can follow instructions and synthesize actions and procedures in natural language. The two initial task domains are artificial agents in simulated robotics and card games. Specifically, for the card games domain, the goal was to develop a system that is able to understand published Solitaire game descriptions in order to play the game. For this Solitaire task we implemented a reference resolution method that, like humans, does not need very complex inferences.

The structure of this chapter is the following: Section 2 gives a brief introduction to the Embodied Construction Grammar formalism. Sections 3 and 4 present the core components: the ontology and the grammar. Section 5 explains the reference resolution method with some explanatory examples and Section 6 describes some more recent work involving ontology driven analysis of Noun-Noun compounds and extensions to metaphorical constructions. Section 7 contains the general conclusions and some ideas for future work.

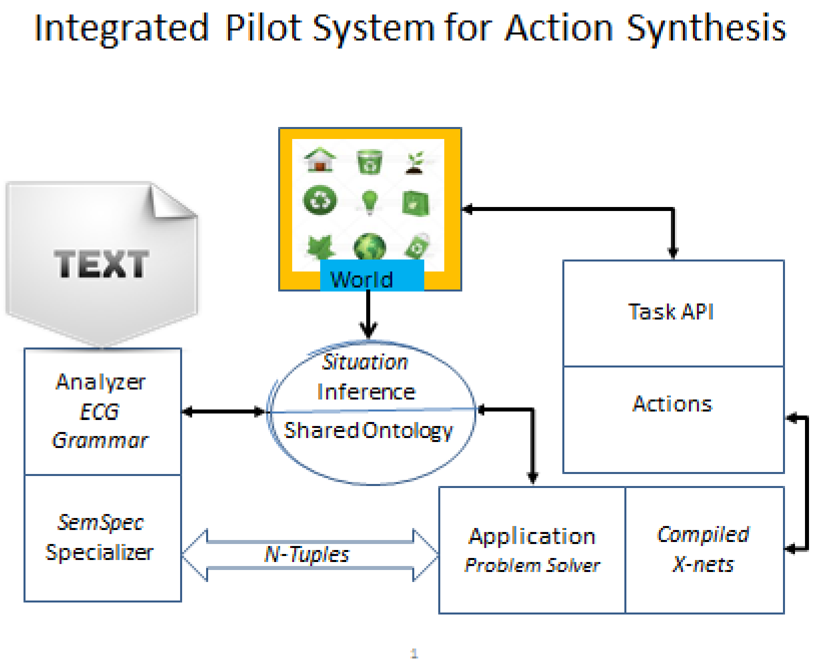


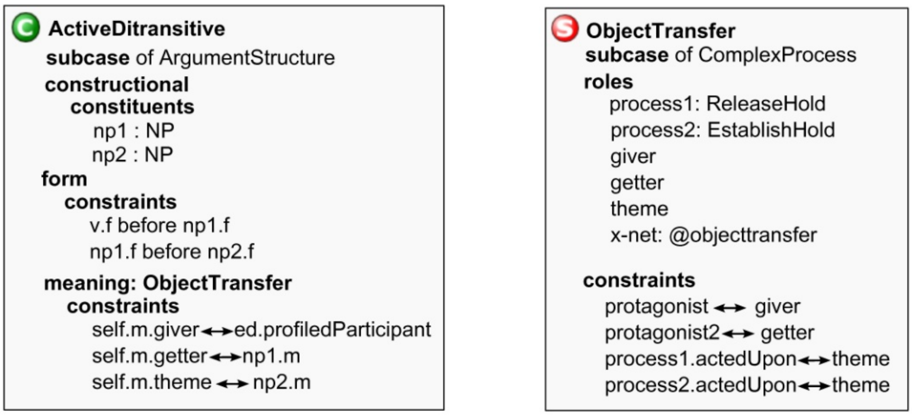
Figure 2: Global system architecture.

2. Embodied Construction Grammar

Embodied Construction Grammar is a formalism for representing linguistic knowledge in the form of construction-based grammars that supports embodied models of language understanding, see (Feldman et al., 2010) for a more extensive review of ECG). ECG is the result of decades of effort by the ICSI/UC Berkeley NTL group to give rise to a formalism that incorporates many insights from cognitive science and construction-based theories of language and covers many empirical findings from neuroscience, linguistics, psychology and computational sciences.

There are two main components of construction grammars: schemas and constructions. Schemas are the basic unit of meaning while constructions represent mappings between form and meaning. Schemas are formed by a list of components (roles) and the different constraints and bindings between these roles. Constructions have several constituents, and represent the form-meaning pairing with the corresponding bindings between the different constituents and the roles of their meaning schemas. Finally, schemas and constructions are not defined in isolation. They are hierarchically structured by *is-a* relations, supporting inheritance semantics along with multiple inheritance. The lattices for schemas and constructions is augmented by an external ontology lattice (cf. Figure 2) that also serves as the terminology bridge to applications, such as Solitaire or robotics.

Figure 3 shows an example of ECG constructions and schemas. The *ActiveDitransitive* construction has three constituents, which are two NPs and a Verb, v, (inherited from the ArgumentStructure construction by the subcase relation). The form block shows the ordering constraints among the constituents of the



```
ActiveDitransitive
  subcase of ArgumentStructure
  constructional
    constituents
      np1 : NP
      np2 : NP
  form
    constraints
      v.f before np1.f
      np1.f before np2.f
  meaning: ObjectTransfer
    constraints
      self.m.giver↔ed.profiledParticipant
      self.m.getter↔np1.m
      self.m.theme ↔ np2.m
```

```
ObjectTransfer
  subcase of ComplexProcess
  roles
    process1: ReleaseHold
    process2: EstablishHold
    giver
    getter
    theme
    x-net: @objecttransfer

  constraints
    protagonist ↔ giver
    protagonist2↔ getter
    process1.actedUpon↔theme
    process2.actedUpon↔theme
```

Figure 3: Example of ECG constructions and schemas.

construction. In our case, it states that the constituent *v* must appear in the sentence before *np1*, and *np1* before *np2*. The meaning of this construction is an *ObjectTransfer* schema, which is a subcase of *ComplexProcess* and has the roles shown on the right in Figure 3. Constructions include a *meaning constraints* block that imposes some bindings between the different constituents of the construction and the roles in its meaning schema. In this case, the *giver* is the *profiled participant* of the event and the *getter* and the *theme* are identified with the meaning of the first noun phrase (*np1*) and the second noun phrase (*np2*) respectively.

In addition to schemas and constructions, the ECG formalism makes use of an ontology that comprises general knowledge about the particular entities present in the discourse. As usual, the ontology is also hierarchically structured allowing multiple inheritance between its elements. We expand the typical entity based ontology with a lattice of functional features that are domain dependent. We discuss the ontology in the following section.

Using these structures, the Analyzer program (Bryant, 2008; Bryant and Gilardi, 2013) produces a deep semantic representation (SemSpec) of the given sentences. The ECG analyzer uses the best-fit score, a metric using a combination of syntactic, semantic, and contextual factors, in order to produce the SemSpecs. Semantic specifications are graphs formed by the bindings and unifications of the ontology items and schemas found in the meaning poles of the recognized constructions. The Semspec captures the semantic and pragmatic information present in the input. SemSpecs are used in the simulation process in the ECG framework (cf. Figure 2). This simulation process is specified by the x-nets (executing networks) which model events and their aspectual structure (Narayanan 1997).

Some previous work has been done on reference resolution within the ECG formalism. For example, Chang and Mok (2006); Mok (2009) present a structured, dynamic context model incorporated in an ECG system for modeling child language learning. This context model is represented using ECG schemas in order to exploit the best-fit mechanisms of the analyzer. In this case, the majority of the workload of the reference resolution method is done by the analyzer using grammatical features (such as number, gender or case) and the ontological categories (when known) of the referent and the anaphor. The resolution process finds the possible antecedents that match the constraints imposed by the referent. This is a very shallow reference resolution mechanism, see (Poon and Domingos, 2008) for a more complex best-fit reference resolution method), with some drawbacks and limitations such as the small number of possible anaphors and antecedents considered by the method and the limited set of features.

3. Functional and entity ontological lattices

Any ontology comprises, in a hierarchical structure, general knowledge about entities and concepts present in the discourse. Our approach expands this general knowledge about entities and concepts by taking into account the functional and other deep semantic properties of those concepts. These properties are often domain dependent since the functionalities of each entity can differ depending on the domain in which they are used. For example, a column has totally different functionalities in architecture than in solitaire games. Thus, the ontology has two main functions. It captures general knowledge about entities, concepts and their functions and also it stores other facts related to the different senses of each of its elements depending on the particular domain, e.g. differentiating between a solitaire game column and an architectural column. The ontology is used by the system to constrain the roles in meaning schemas of the recognized constructions, reducing the possible candidate schemas and helping the analysis process. This will be even more important in the examples of Section 6.

```
(type entity sub item)
(type function sub item)  // top of function lattice
        (type container sub function)
                  (type container-sol  sub container)
                            (type column-sol sub entity container-sol  )
                            (type tableau-sol sub entity container-sol  )
        (type moveable sub function)
                  (type moveable-sol sub moveable)
                            (type card sub entity moveable-sol )
                                      (type king-sol sub card)
                                      (type ace-sol sub card)
```

Figure 4: Fragment of the entity and functional ontology sub-lattices.
The two reserved words are **type** and **sub**, indenting is for ease of reading.

The ontology used in the Solitaire domain is structured by two connected inheritance sub-lattices (see Figure 4). All of the intra- and inter-lattice relations support the usual inheritance semantics including multiple inheritance. The Entity sub-lattice entities are structured hierarchically following the usual *is-a* relations in order to specify the categories to which each of its elements belongs. The Functional sub-lattice of the ontology is a hierarchical structure of functions and properties of the elements in the *entity sub-lattice*. Notice, e.g., that **card** is a subcase of both **entity** and **moveable-sol**, and is thus present in both sub-lattices.

Another aspect of the ontology is the domain-specific information. This information is needed to distinguish the different senses of each word. For example, a column in the solitaire domain has the function of *container*. However, a column in an architectural domain does not have that function. And the same is applicable to the functional lattice: the *container* or *moveable* functions can have different interpretations depending on the domain. So the different senses of each word have to be related to a specific domain (in Figure 4, for example, *column-sol* inherits from *container-sol*).

4. Schemas, constructions and ontological constraints

As stated in the introduction, we avoid complex reference resolution algorithms by using an appropriate design of the grammar constructions and schemas and constraining some roles to the functional categories described in the previous section.

Solitaire play rules usually describe card movements. Typical sentences involve a trajector (a card, or rather, any *moveable* entity) and a landmark (a position in the tableau, or rather, any *container*). See, for example, sentences like: “move aces to foundation spaces”, “put cards on the waste” or “fill tableau spaces with kings”. Given that all these sentences have the meaning of moving something moveable to a container, we designed a schema called *Update-sol* to express their meanings. That schema requires that the landmark should have the ontological feature of being a *container-sol* and the trajector should be *moveable-sol*. It is important to note that the *Update-sol* schema constrains those roles to functional categories and not to entity categories. In a simple solitaire game, we could think that only cards can be moved so we could constrain the *trajector* role to the *card* ontological category. However, in general, we can move many other things such as groups of cards, columns or piles. Thus, the role restrictions should not be constrained to concrete entities but also to functional categories. This is a paradigm example of the semantics-based constraint satisfaction methodology that is the central idea of this chapter.

In order to correctly analyze the instruction sentences, different argument structure constructions were built. All of them have in common that they have the *Update-sol* schema as their meaning pole. Each of the constituents of those argument structure constructions is bound to the corresponding role of the *Update-sol* schema. Therefore, the constituents inherit the functional constraints imposed by the *Update-sol* schema. This point is of crucial importance in the reference resolution method proposed here. The problem with many naïve methods is that they only look for compatibility of grammatical features and of *ontological* categories. However, there exist some undefined anaphors that do not have an ontological category (for example, pronouns like *it*). The ECG formalism allows us to define the functional categories of all the elements in a given argument structure. This way, although we can not state the entity-ontological category of pronouns, we can know their *functional* categories (i.e. we can know whether *it* should be a container or should be moveable in this context).

As we will see below, this distinction is of great importance in our reference resolution method.

Figure 5 shows one of the constructions used in the analysis of the sentence: “Fill tableau spaces with kings”. The *Update-sol* schema transmits the functional constraints to the constituents of the argument structure. This way, we impose that the *patient* constituent of a *Fill-with* argument structure has to be a *container* and that the np of the adverbial phrase has to be *moveable*.

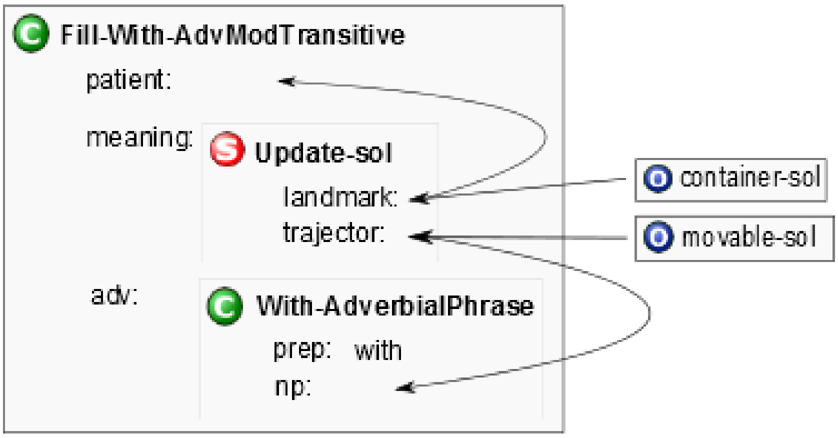


Figure 5: Fragment of the constructions used to analyze sentences like: “Fill tableau spaces with kings”.

5. Ontology driven reference resolution

Reference resolution is driven by a variety of constraints. Much work has been done on how best to combine those constraints in many sophisticated ways. We will focus only on two specific constraints, plus recency. Syntactic constraints include agreement compatibility (i.e. number, gender or case compatibility) and semantic constraints specify functional compatibility. Given the constructions and ontological structures presented in the two previous sections, our reference resolution process is quite simple. We exploited the probabilistic features of the best-fit analysis included in the ECG framework (Bryant 2008) which restricts possibilities and based our method on the search of possible anaphors and antecedents and the comparison of their syntactic and semantic features.

As stated before, the analyzer builds a semantic specification as the meaning representation of the given sentence. A semantic specification is basically a network of the constructions that fit the structure of the analyzed sentence and the schemas and ontology items that fill the meaning poles of those constructions. Our resolution algorithm is based on syntactic and semantic compatibility between the anaphor and the antecedent and the recency principle. For each anaphor, it selects the most recent antecedent that shares its syntactic features (basically agreement features such as number, gender or case) and its semantic features (the functional roles) with the anaphor.

The resolution module goes through the semantic specifications produced by the analyzer and keeps track of the possible antecedents it finds. Possible antecedents could be proper nouns or noun phrases in general. A list of the possible antecedents and their syntactic and semantic features is stored in inverse order (so the first element is the most recent possible antecedent). Once the module finds a possible anaphor (which is usually a pronoun or a definite noun phrase like “those cards”), it tries to resolve it. Given an anaphor, it goes through the list of possible antecedents and tries to find one with the same syntactic features (number, gender) and the same functional category. The method returns the first antecedent candidate that matches. In other words, it chooses the most recent antecedent which shares the grammatical features and functional categories with the anaphor.

Notice that a given noun phrase could be, at the same time, an anaphor and an antecedent. For example, in these sentences:

a) *Kings can be moved to an empty space.*
b) *Those cards can also be moved to a foundation space.*
c) *Moreover, they can be moved to another column.*

The noun phrase “those cards” in sentence b) is an anaphor for “Kings” and the antecedent of “they” in sentence c). Therefore, when our method finds a noun phrase, it applies the two mechanisms: store the noun phrase as a possible antecedent and, if it is anaphoric, try to resolve it with the previous possible antecedents. This way, our method would find that “kings” is the antecedent of “those cards” and “those cards” is the antecedent of “they”. Therefore, our method also establishes a link between “they” and “Kings” to capture the rule that “Kings” can be moved to another column.

In order to gain a better understanding of how our method works, we will walk through an example. Suppose that the following two sentences are found in a Solitaire game description.

- *Move the top card of a column to the top of another column.*
- *Move it to a foundation pile.*

As mentioned in the previous section, *move*, requires that its trajector must have the functional category of being moveable. In constructional terms, the *move* construction binds (unifies) its trajector role with the functional category *moveable-sol*. When the resolution binding is made, the pronoun *it*, which has no ontological category by default, takes the *moveable-sol* category through unification. Then, the system goes through the list of possible antecedents comparing the syntactic and functional features. The first element would be *column*, whose syntactic features match up with the ones of the pronoun *it* (in this case, we just have to compare the grammatical number of the anaphor and the antecedent). However, their functional categories do not match since *column* has the *container-sol* functional category. Thus, the system continues to the next possible antecedent in the list, which is “*the top card of a column*”. In this case, the syntactic and functional categories are the same so the system establishes that *it* refers to “*the top card of a column*”. It is important to note that the inclusion of the semantic constraints in the reference resolution algorithm is crucial. A simple reference resolution method based only on the recency principle and syntactic compatibility would say that “column” is the antecedent for “it” in our example. However, the ECG formalism, allows us to require that “it” should be something moveable and therefore, can not be bound to “column”.

This simple reference resolution method has been tested on a small solitaire domain with remarkably good performance. Obviously, it could be improved in many ways, but the use of deep semantics and domain-specific functional categories appears to be a generally powerful tool in reference resolution. Current work is focused on more general semantic constraints such as affordances and materials and ontological categories such as person and institution.

6. More applications of ontology-based constraint satisfaction

There are, of course, many challenging problems in natural language understanding and no single method will be effective for all of them. However, we have found that deep ontology-based constraint satisfaction has been helpful in a much larger range of tasks than we originally envisioned. In this section, we will try to demonstrate how and why this seems to work.

Any grammar involves constraints and any Analysis system (or parser) uses constraint satisfaction, either absolute or weighted, in the sense that a grammar specifies the constraints on a parse. The key insight is that many of the required constraints are at a rather deep conceptual level. It turns out that Embodied Construction Grammar has two properties that are enormously helpful: constructions as form-meaning pairs and meaning as deep semantic schemas. In the Solitaire example above, the crucial deep meaning concerned whether a construct denoted an entity in the game that was moveable or was a possible destination. Constructions that require such constraints on its arguments are simple and natural, given that there is an ontology lattice of such properties. We will next illustrate how very similar reasoning produces an effective treatment of many English noun-noun compounds and related constructions.

In general, it is impossible to understand all noun-noun compounds out of context. For example, the written form “steel box” could mean box made of steel, a box containing steel, or various other things in special contexts. Notice also that “box of steel” has the same basic ambiguities. And there are cases like “pear bus” that have no obvious interpretation out of context. However, if two buses of children were on a trip and only one was stopping at a pear orchard, the compound above is quite natural.

To understand how the best fit of noun-noun compounds works, we will need to delve further into the ECG treatment of noun phrases, whose meaning is captured in the ReferentDescriptor (RD) Schema, which is shown in Figure 6, along with two related schemas.

```
schema RD // meaning of NP
roles
  ontological-category
  givenness: @givennessValues
  referent
  number: @RDnumberValues
  gender: @genderValues
  bounding: @boundingValues
  scale: @scale
  amount: Amount // various schemas for quantifiers, etc.
  extensions: Extensions // various schemas for affordances

schema Artifact
subcase of Extensions
roles
  affordances: RD
  material: RD
constraints
  material.ontological-category <-- @material

schema Container
subcase of Artifact
roles
  contents: RD
constraints
  contents.bounding <-- @indeterminate //mass or plural
```

Figure 6. ECG semantics of Noun Phrases

Several RD roles are important for our examples. Most basic is the `ontological-category` role, which unsurprisingly has values from the Ontology, partially depicted in Figure 7 below. For example, the lexical noun, *block*, has @block as its `ontological-category` which is a subcase of artifact in the ontology lattice, as shown. The traditional `number, gender, givenness, and bounding` roles play their usual roles in constraining analysis, but are not needed for our examples. The last role, `extensions,` is central to the current set of examples. The bottom of Figure 6 shows two particular extension schemas: Artifact and Container. The ECG extension schemas, like `Artifact,` are used to capture the role bindings for extension types. Consider the `Artifact` schema. We will not use the *`affordances`* role here, but it can capture functional properties like *`moveable`* in the Solitaire example. The *`material`* role captures the fact that an artifact is made of some (main) material and the RD describing that substance is constrained to be of ontological type @material. As we can see from Figure 7, *beer* is a mass, but not a material in the ontology. Looking ahead, this constraint will lead the analyzer to construe “beer box” as a box containing beer, but not as a box made of beer.

```
(type entity sub item)
        (type shaped sub entity)  // contrast with mass
                   (type socialEntity sub shaped)  // count
           (type institution sub socialEntity  )
       (type artifact sub shaped)
                   (type block sub artifact)
                      (type container sub artifact)
                          (type box sub container)
         (type mass sub entity)  // contrast with shaped
              (type discreteAble sub mass)  //  “one beer”
                  (type beer sub discreteAble)
                (type material sub mass)
                    (type steel sub material)
 //  metaphor example
(type animate sub shaped moveable physicalEntity)
```

```
    (type person sub animate sentient)
(type institution sub socialEntity  )
    (type city-inst sub institution )   // 2 conceptualizations of city
(type metaphors sub sharedEnumeration)  // names of known metaphors
    (type institutionAsPerson sub metaphors )
```

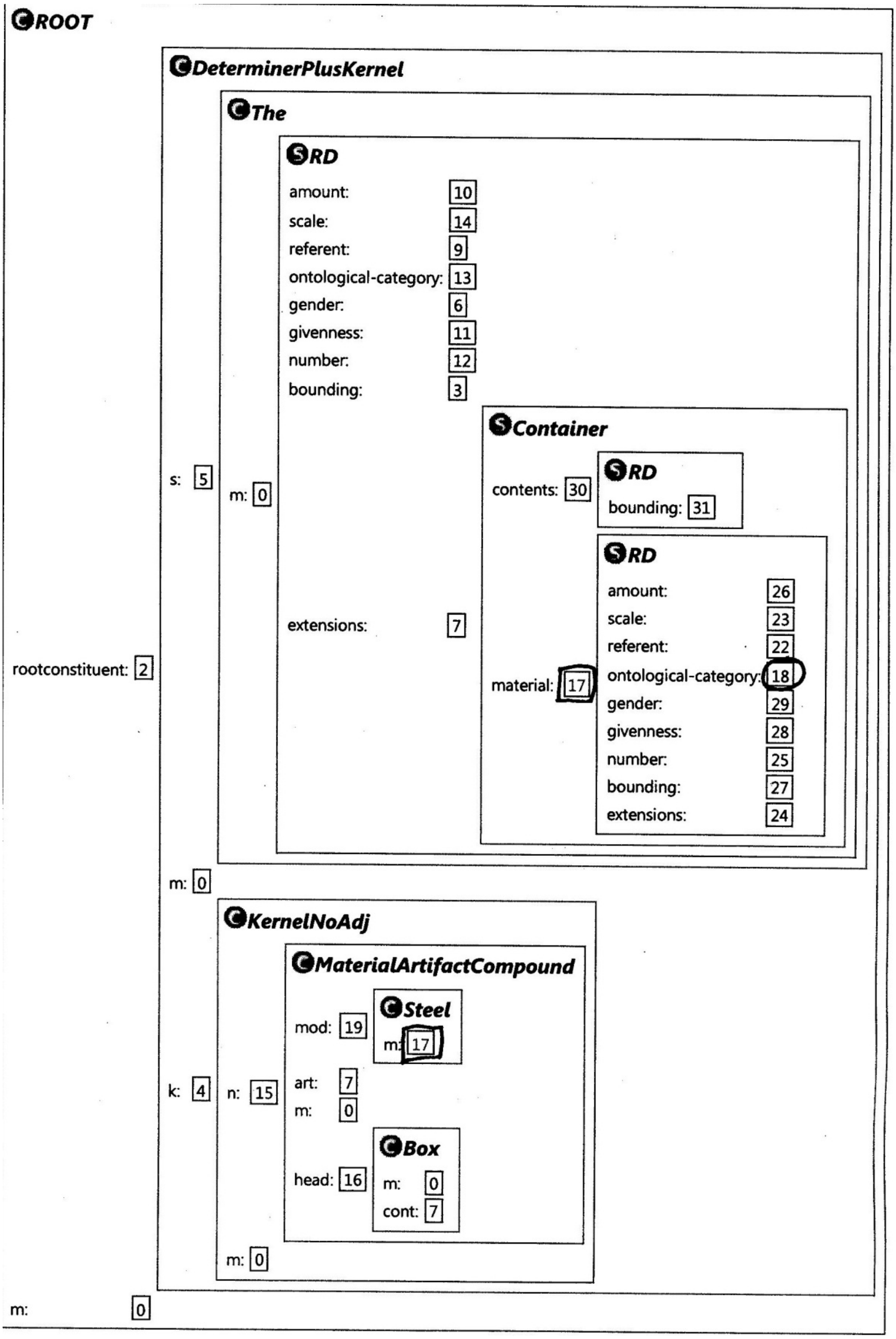
Figure 7. Ontology Fragment for our examples

To further understand our examples, we examine one crucial `construction: MaterialArtifactCompound,` shown in Figure 8. This is more complex than the construction examples shown in Figure 3, but is similar in structure. The two noun constituents are a head and a mod(ifier) with the obvious form constraint. For this construction to match, the `ontological-category` of the head role must be `@artifact` and that of the mod role must be `@material` in the ontology (Figure 7). This construction will match “steel box”, but not “beer box” The other three lines in the meaning block of Figure 8 specify the required bindings. It is interesting to look at the relation between this construction and the `Artifact` schema of Figure 6. The `Artifact` schema requires that the `ontological-category` of its `material` role be the ontological item `@material.` The construction of Figure 8 requires the same of its `mod` constituent. This constraint is satisfied for materials like steel, but not for more general mass entities like beer, as shown in the ontology fragment of Figure 7. Also, the `art.material <--> mod.m` constraint binds the specific material named by the mod constituent to the material role of the art extension of the final RD. Thus the resulting SemSpec will specify what the artifact is made of (cf. Figure 9).

A full `MaterialArtifactCompound` analysis of “the steel box”, including semantic bindings, is given as Figure 9. The ECG SemSpec display uses the standard boxed-number notation for bindings. For example, in Figure 9, the boxed number 17 captures the binding between the material of the Container and the meaning of the noun “Steel”. Also, the boxed number 18 is bound to the ontology item @steel, although ontology items are currently a separate display.

```
construction MaterialArtifactCompound
 subcase of Nominal
constructional
      constituents
        mod: Nominal
        head: Nominal
constraints
      self.features <--> head.features
form
constraints
      mod.f meets head.f
meaning
evokes Artifact as art
constraints
      head.m.ontological-category <-- @artifact
      mod.m.ontological-category <-- @material
      art.material  <--> mod.m  //sets value directly
      self.m.extensions <--> art
      self.m <--> head.m
```

Figure 8. The `MaterialArtifactCompound` Construction



ROOT

rootconstituent: 2

DeterminerPlusKernel

s: 5

The

m: 0

RD

amount: 10

scale: 14

referent: 9

ontological-category: 13

gender: 6

givenness: 11

number: 12

bounding: 3

extensions: 7

Container

contents: 30

RD

bounding: 31

material: 17

RD

amount: 26

scale: 23

referent: 22

ontological-category: 18

gender: 29

givenness: 28

number: 25

bounding: 27

extensions: 24

m: 0

k: 4

KernelNoAdj

n: 15

MaterialArtifactCompound

mod: 19

Steel

m: 17

art: 7

m: 0

head: 16

Box

m: 0

cont: 7

m: 0

m: 0

Figure 9. SemSpec for “the steel box” as a `MaterialArtifactCompound`

The current grammar also includes a construction for analyzing N-N compounds like “beer block” that have no obvious interpretation to the system or to us. This is the first example in Figure 10 below. The line marked ***** indicates that the system has only the general analysis of this phrase. By contrast, "the steel box" has two additional readings with respect to the current grammar. The marked lines illustrate the readings for the box made of steel and the box containing steel, respectively.

```
"the beer block"
        ROOT[1] (0, 3)
        DeterminerPlusKernel[0] (0, 3)
        The[6] (0, 1)
        KernelNoAdj[4] (1, 3)
        NounNounCompound[15] (1, 3)  *****
        Beer[16] (1, 2)
        Block[18] (2, 3)

"the steel box"
        ROOT[0] (0, 3)
        DeterminerPlusKernel[1] (0, 3)
        The[12] (0, 1)
        KernelNoAdj[8] (1, 3)
        MaterialArtifactCompound[15] (1, 3) )  *****
        Steel[18] (1, 2)
        Box[16] (2, 3)

        ROOT[1] (0, 3)
        DeterminerPlusKernel[0] (0, 3)
        The[6] (0, 1)
        KernelNoAdj[4] (1, 3)
        ContainerContentsCompound[15] (1, 3) *****
        Steel[20] (1, 2)
        Box[17] (2, 3)
```

Figure 10. Constructional analysis of more N-N compounds

Metaphor and Constraint Satisfaction

The same general approach using deep semantic constraints can handle a wide range of metaphorical constructions and this is one of our major current efforts. The last 6 lines of the ontology fragment in Figure 7 provide a basis for one introductory metaphor example. The first two of these lines depict part of the ontology lattice defining @person as a subcase of @animate and @sentient. The next two lines depict how a city as an institution (@city-inst) is ontologically a subcase of @institution. The final two lines of Figure 7 indicate how word-level metaphors, such as @institutionAsPerson can be incorporated directly into the ontology.

Now consider a metaphorical noun phrase “"the desperate city". An emotional attitude can not normally be associated with an institution, and this constraint is similar to the ones we have been discussing. The DeterminerPlusKernel construction incorporates a deep semantic constraint between the adjective (ap) and head noun (n):

```
ap.m.domain <--> n.m.ontological-category
```

For non-metaphorical noun phrases such as “the desperate man”, the head noun directly meets this ontological constraint. However, the grammar also includes the metaphor **institutionAsPerson,** which captures the fact that an institution can be conceptualized as a person. More specifically, the KernelInstAsPerson construction utilizes this metaphor, along with the constraints that allow a head noun with ontological-category @institution to be modified by an adjective whose domain is @person. Since the ontological-category for the lexical noun *city* is a subcase of @institution, it meets the more general constraint on the head noun of this construction.

As we can see from Figure 11, the best fit analyzer accepts the phrase “"the desperate city" and explicitly includes a metaphor instance in the resulting SemSpec. Current efforts extend this paradigm to a much wider range of semantics (e.g., scales) and metaphorical constructions. For example, the system can now analyze complex examples like “the desperate government moved to the left”

```
"the desperate city"
        ROOT[1] (0, 3)
        DeterminerPlusKernel[2] (0, 3)
        The[5] (0, 1)
        KernelInstAsPerson[4] (1, 3)  ******
        Desperate[16] (1, 2)
        City-inst[19] (2, 3)
```

Figure 11. Metaphorical analysis of "the desperate city"

7. Conclusions and future work

In this chapter we have described how a deep semantic ontology can help solve difficult problems in language analysis using Embodied Construction Grammar (ECG). The main example was a reference resolution mechanism and its application to the understanding of card games instructions. The method exploits the features of the ECG formalism and the best-fit analyzer in order to avoid complicated reference resolution approaches. Within this framework we built a method that, like humans, does not need very complex inferences in order to solve the basic reference resolution task. The method was based on the recency principle and grammatical and conceptual compatibility between the anaphor and the antecedent.

In order to check the compatibility, we used standard agreement features (such as number, gender or case) and we also introduced some conceptual features (such as the different functionalities of the concepts referred by the anaphor and the antecedent) in order to improve the performance of the method. All this yielded an efficient and also accurate method that has been tested as a part of a prototype system that tries to understand solitaire game descriptions well enough to play the game. Referring back to Figure 2, the reference resolution techniques described above are part of the *Specializer*, shown in lower left of the Figure. This prototype system was not continued because the language of card games is too specialized to be a good platform for general language understanding research.

More recent work is focusing on two domains: instructions for robots and deep analysis of metaphorical language. Both of these task domains can be seen as instances of the general architecture depicted in Figure 2. In the robotics domain, the research focus is on being able to instruct a robot on how to do complex tasks. This involves a tractable and powerful mode of actions (Narayanan 1997) and ECG grammar for language about actions. Since, language almost always under-specifies actions, there is a need for using goals and context to complete the specification. We expect the rich semantic ontology to play an important role in this Specifier task.

For the metaphor understanding project, the role of the deep ontology is even more direct. It is now clear that much of metaphor recognition and use depends on mapping between conceptually divergent source and target domains. The constraint satisfaction methodology described in this chapter has already proven to be extremely useful, as indicated in the example of Figure 11.

These results, along with earlier ECG uses of constraint satisfaction methods, are promising as part of systems for language understanding based on deep, embodied semantics. Referring back to Figure 2, for any Application, the role of the Specializer is to convert the general semantic analysis given as the SemSpec into task specific N-tuples that convey the information needed by the Application. This obviously includes determining any references resolvable by discourse and situational context, as discussed in this chapter. Unfortunately, the Specializer itself is the one module of the system that we have not been able to specify non-procedurally and we view this as a major open problem for the constraint satisfaction approach to language understanding.

Acknowledgements

This work was partly funded by a JAE Predoctoral Fellowship from CSIC (Spain). We are thankful to the International Computer Science Institute and the University of California at Berkeley and, in particular, the people on the Neural Theory of Language project. The reviewers provided many useful suggestions.

6. References

Asher, N., Lascarides, A., 2003. Logics of conversation. Cambridge University Press, Cambridge, UK.

Bryant, J., 2008. Best-Fit constructional analysis. Ph.D. thesis. UC Berkeley. Berkeley,CA.

Bryant, J., Gilardi, L., 2013. A cognitive model of sentence interpretation, in: HC, Boas. (Ed.), Computational approaches to embodied construction grammar, John Benjamins, San Diego.

Chang, N., Mok, E., 2006. A structured context model for grammar learning, in: Proceedings of the 2006 International Joint Conference on Neural Networks, Vancouver, BC.

Feldman, J., 2006. From Molecule to Metaphor: A Neural Theory of Language. MIT Press.

Feldman, J., Dodge, E., Bryant, J., 2010. Embodied construction grammar, in: Heine B, N.H. (Ed.), The Oxford handbook of linguistic analysis, Oxford University Press, Oxford. pp. 111 – 138.

Hobbs, J., 1976. Pronoun resolution. Technical Report. CUNY.

Hobbs, J., 1978. Resolving pronoun references. Lingua 44, 311–338.

Kehler, A., 2002. Coherence, reference and the theory of grammar. CSLI Publications, Stanford, CA.

Lappin, S., Leass, H., 1994. An algorithm for pronominal anaphora resolution. Computational Linguistics 4, 535–561.

Mitkov, R., 1998. Robust pronoun resolution with limited knowledge, in: Proc. COLIN/ACL'98, pp. 869–875.

Mok, E., 2009. Contextual bootstrapping for grammar learning. Ph.D. thesis. UC Berkeley. Berkeley,CA.

Narayanan, S., 1997. KARMA: Knowledge-based action representations for metaphor and aspect. Ph.D. thesis. UC Berkeley. Berkeley, CA.

Ng, V., 2007. Semantic class induction and coreference resolution, in: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL-07).

Ng, V., Cardie, C., 2002. Improving machine learning approaches to coreference resolution, in: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pp. 104 –111.

Ponzetto, S., Strube, M., 2006. Exploiting semantic role labeling, wordnet and wikipedia for coreference resolution, in: Proceedings of the HLT 2006, New York City. pp. 192 – 199.

Poon, H., Domingos, P., 2008. Joint unsupervised coreference resolution with markov logic, in: Proceedings of the 2008 conference on empirical methods in natural language processing (EMNLP), Honolulu, Hawaii. pp. 650–659.

Prince, Alan & Smolensky, Paul. 2004. *Optimality Theory: Constraint interaction in generative grammar.* Blackwell.

Soon, W., Ng, H., Lim, D., 2001. A machine learning approach to coreference resolution of noun phrases. Computational Linguistics 4, 521 – 544.