

METAPHORIC GENERALIZATION THROUGH SORT COERCION

Ellen Hays
10 Pine Avenue
Arlington, MA 02174
hays@linc.cis.upenn.edu

Samuel Bayer
The MITRE Corporation, A040
Burlington Rd.
Bedford, MA 01730
sam@mitre.org

Abstract

This paper presents a method for interpreting metaphoric language in the context of a portable natural language interface. The method licenses metaphoric uses via coercions between incompatible ontological sorts. The machinery allows both previously-known and unexpected metaphoric uses to be correctly interpreted and evaluated with respect to the backend expert system.

1 Introduction

One of the central issues in AI systems has been how to model the domain: what are the primitives of the ontological language, how are the ontological sorts organized, and so on. AI researchers have explored a wide range of object-centered and relation-centered representations (for example, Brachman and Schmolze (1985) and Minsky (1975)). When setting up the domain model for a natural language interface, though, one must also keep the lexicon in mind, so that words can be defined and processed efficiently; if possible, the hierarchical organization of the domain model should minimize sense ambiguity, by allowing lexical items to point to classes that dominate the objects that reflect each item's range of meanings.

However, a growing body of literature argues that the generalizations about the world implied by the lexicon do not correspond exactly to standard computational notions of fine-grained ontological structure. Rather, the mapping is mediated by pervasive low-level metaphoric and metonymic processes (as pointed out by Lakoff (1987) and others) that make for a mismatch between the desired world model and the lexicon.

At the MITRE Corporation, we are developing an interface architecture to support King Kong, our portable natural language interface for expert systems, and AIMI, our multimedia interface for the same class of systems.¹ Portable interfaces provide an additional set of problems beyond simple domain modeling. In particular, in our case, the structure the knowledge representation imposes on the backend domain model is hierarchical and relation-based, and its form must be consistent across system ports; thus the knowledge representation may structure domain-specific information in a way that is fundamentally different from the way it is organized in the backend. In this context, one needs to develop a computational account of the low-level metaphor that creates the mismatch between the domain model and the lexicon. In this paper, we will discuss a mechanism implemented in King Kong that we call "sort coercion" that is intended to address that mismatch.

2 Refinement in the King Kong domain model

In the King Kong knowledge representation, both concepts and relations are organized hierarchically. King Kong exploits this hierarchy in a number of ways, of which the most relevant to this discussion occurs in the process of *refinement*.

When King Kong interprets a sentence, it builds an *interpretation* corresponding to the input. Interpretations represent a point in the semantic

¹The AIMI system is, in fact, one of the domains to which King Kong has been ported. The current implementation of King Kong has also been ported to two mission planning systems and one transportation planning system. The coercion mechanism described here currently supports examples in the mission planning and interface domains.

analysis that is subsequent to some lexical disambiguation but prior to the determination of scope relationships and reference resolution. They are built in large part out of knowledge representation objects. They have heads, for instance, which are typically filled by relations from the domain model, and argument lists, which are usually mappings from the arguments of the relation in the head to other interpretations.

The heads of these interpretations can be very general relations, and King Kong uses refinement to find relations in the hierarchy that are dominated by the head indicated by the input and that are specific enough to be evaluated. Once referents have been resolved, refinement chooses appropriate leaf relations by recursively checking the children of each relation in the subgraph accessible from the input relation and eliminating any children whose argument restrictions are disjoint from the sorts of the arguments. Each leaf relation has *backend access code* stored on it that allows King Kong to communicate with the backend expert system. The code stored on the leaf relations found by this procedure supports the evaluation of the logical expressions generated from the input interpretations.

3 Motivations for sort coercion

The obvious problem for a system using a hierarchy of the kind just described is that in most cases there is no direct, one-to-one mapping between words and concepts. Most lexical items have a number of different meanings, and within those meanings there are often different senses, as well as various selectional restrictions and preferences, whether rigidly defined or merely stylistic.

One case in point is the locative prepositions, which have been studied in great detail by a number of linguists, including Herskovits (1986), whose analysis of static locative prepositions such as *in*, *on*, and *at* defines a program of sorts for interpreting each, in the presence of particular arguments. The scheme consists of an *ideal meaning* (a very abstract definition) and a number of *use types* (more concrete senses). The relations so defined, however, require that the system have recourse to a number of "functions" that, in some sense, "coerce" the objects arguments to the relations from one ontological sort to another.

Herskovits calls these *geometric description functions*; they capture a number of different kinds

of conceptualization (or recasting) of objects. For example, for the purposes of the abstract relation $at(x, y)$ ("X [is] at Y"),² both x and y are taken to be points.³ Then in the actual instance of the relation $at(john, airport)$, according to this model, we have conceptualized both of the (three-dimensional) objects in the relation as points in order to express that particular locative relation between them. In the same way, when we use *at* with a temporal argument ("a meeting at 5 o'clock"), we are in some sense "viewing" a time point as a spatial object, namely a geometric point.⁴

Since a geometric description function can apply to any argument of the appropriate ontological sort (i.e., within the range of the function), regardless of the relation it figures in, what this scheme captures is a generalization about conceptual "transfer of reference", as Herskovits has more recently called it (Herskovits, 1989).

The coercion mechanism described in this paper was inspired partly by Herskovits' work and partly by the system's existing domain model. It is a response to the need for a one-to-many mapping from lexical items to ontological items (in this case locative and event relations), and is an attempt to capture explicitly some of the ways in which changing the way an object is viewed allows certain metaphoric and metonymic uses.

4 The coercion mechanism

The central information source in our account of metaphor and metonymy is a set of *coercion rules*. Coercion rules declare different ways of viewing particular classes of objects. So if we wish to view temporal intervals as one-dimensional spatial objects (lines), we would declare:

(1) (`defCoerce temporal-interval line`)

These coercion rules can be chained; if we wish to view events as temporal intervals (that is, the intervals over which they occur), we could ultimately view them as lines as well simply by adding another declaration:

²Herskovits follows Talmy (1983) and others in seeing locative prepositions as defining a figure/ground relationship between a located object and a reference object.

³The ideal meaning of *at* is for two points to coincide (1986, p.128).

⁴Jackendoff proposes a similar response to the problem, with respect to temporal use of spatial expressions. See (Jackendoff, 1983, ch.10).

```
(2) (defCoerce
      durative-event
      temporal-interval)
```

King Kong uses these coercion rules in two related ways. The first is to license what we call *shadow relations*. These are relations that have no parent but are connected to the domain model by means of a *shadow* link. This link requires that the value restrictions on the arguments of the shadowing relation be connected to the value restrictions on the shadowed relation by a chain of coercion rules. These shadow links are required because the normal subsumption relationship does not permit the shadowed relations to be connected to their shadows; the endpoints of coercion links will typically be disjoint. Intuitively, these shadow relations represent the metaphoric uses that Lakoff called attention to. When King Kong encounters a relation pointed to by the input that has shadows associated with it, it exploits an expanded version of the refinement mechanism described in Section 2 to search through not only children but also shadows for acceptable leaf relations.

Let us take a brief example. Imagine that we wish to capture the low-level metaphor in a sentence like "The length of the meeting is 5 hours." The ideal meaning of the *length-of* relation involves a line and a one-dimensional (spatial) measure, which are the value restrictions on the two arguments (indicated here as *vr*):

```
(3) (defRelation length-of
      (arg object (vr line))
      (arg measure (vr id-measure))
      (super measure-of))
```

The coercions described in (1) and (2), together with a view of quantities of time as spatial measures (shown in (4)), suffice to license the shadow embodying the temporal metaphoric use of the *length-of* relation in (3):

```
(4) (defCoerce
      quantity-of-time id-measure)
```

```
(5) (defRelation length-of-event
      (arg event
        (vr durative-event))
      (arg measure
        (vr quantity-of-time))
      (shadows length-of))
```

But the mechanisms introduced so far do not address a particular requirement of the King Kong

metaphor mechanism that might not be imposed on other such mechanisms: the resulting logical expressions must be evaluable. Since King Kong is an interface, its domain model captures the shape of the data, but it does not itself store any facts; it must consult an external (i.e., the backend system's) database to reply to any queries. So when it recognizes a metaphoric use, it must provide the proper backend argument fillers to the backend database in order to evaluate the query. But if the metaphoric use of the relation corresponding to the input has an argument corresponding to *event* and the ideal meaning requires an argument corresponding to *line*, as in the *length-of* relation given above, how can King Kong provide the proper backend individuals?

The answer lies in the way coercion rules interact with the domain model. When they license a shadow relation, they *instantiate* a point in the space of possible coercions, and to this shadow relation we can attach backend access code that expects objects corresponding to the classes in the value restrictions of the current (shadowing) relations. In other words, in the example given above, although conceptually we are viewing an instance of *event* as an instance of *line*, we need not refer to the ideal class at all in processing; the shadow relation permits us to treat these instances as ordinary members of the *event* class. The existence of this shadow implies that there is a conceptual mismatch between the way the backend system records this information and the way language expresses it; the backend system considers the input classes directly, while the ontology and lexicon view these classes as coercions from other classes.⁵

But what if the backend system requires that the input classes be coerced, just as the domain model and lexicon do? This is the second way in which the coercion rules can support metaphoric language. Coercion rules can have fragments of logical expressions attached to them that describe how to convert items of one class to items of another. We can use these augmented coercion rules to process novel uses of relations. If a path of coercions can be followed dynamically (rather than built at load time, as when shadows are licensed), the novel use can be evaluated, as long as the log-

⁵ This shadow, along with many others, could be automatically generated from our set of coercion rules, but since the backend access code that shadows are "repositories" for cannot be automatically generated as well, that would not be productive. Furthermore, we acknowledge the possibility that the unconstrained application of these coercion rules would generate shadow relations with no linguistic validity.

ical expressions attached to the coercion rules can themselves be evaluated. In that case, the procedure that builds logical expressions will fold the logical expressions associated with the coercion rules into the overall logical expression, in order to create an evaluable expression.⁶

For example, consider a backend system that knows about meetings and their start and end times, but doesn't store their duration. Furthermore, it knows how to manipulate intervals of time. We might amend the coercion rule in (2) above in the following way, and replace the shadow shown in (5):

```
(6) (defCoerce
      durative-event temporal-interval
      (lambda x
        (durative-event-has-interval
         durative-event x)))
```

```
(7) (defRelation
      durative-event-has-interval
      (arg event
        (vr durative-event))
      (arg interval
        (vr temporal-interval))
      (super event-has-property))
```

```
(8) (defRelation length-of-interval
      (arg interval
        (vr temporal-interval))
      (arg measure
        (vr quantity-of-time))
      (shadows length-of))
```

In this situation, the `length-of-interval` relation instantiates a point in the space of possible coercions that represents the system's ability to compare a temporal interval with a time measurement. It represents the direct understanding of something like "The length of the coffee break was 10 minutes," where we assume that a coffee break is a kind of temporal interval. Ignoring tense, the logical expression corresponding to this example is:⁷

```
(9) (length-of coffee-break1 10-minutes)
```

The generalized refinement process will locate the shadow `length-of-interval` and use the

⁶If the coercion rules are not all evaluable, we can build an interpretation for the input, but we cannot evaluate it.

⁷King Kong actually represents measurements as undifferentiated pools of individuals, much as it represents "10 planes", for instance. We may ignore that detail here.

code associated with it to communicate with the backend system. We can do more, however. Given the existence of the augmented coercion rule, we can understand sentences like our first example "The length of the meeting is 5 hours" by building a chain of coercions that consists of a single link, from events to temporal intervals. In this case, our logical expression will be:

```
(10) (exists y
       (lambda x
         (durative-event-has-interval
          coffee-break1 x))
       (length-of y 10-minutes))
```

As long as there is backend access code associated with the `durative-event-has-interval` relation, we can process this use of the `length-of` relation without the shadow in (5) (`length-of-event`) present. In fact, we can process *any* metaphoric reference to an event that appears in an argument position whose filler is restricted to intervals of time. Consider the `overlap` relation, whose ideal meaning is a relation between two planes or two lines. The coercion rules already given will license a shadow that relates two intervals:

```
(11) (defRelation overlap
       (arg obj1 (vr line))
       (arg obj2 (vr line))
       (super static-locative))
```

```
(12) (defRelation temporal-overlap
       (arg obj1
         (vr temporal-interval))
       (arg obj2
         (vr temporal-interval))
       (shadows overlap))
```

The shadow in (12) corresponds to an example like "The current calendar year overlaps with the next fiscal year." But given the augmented coercion rule, we can understand sentences like "The first meeting overlaps with the second meeting" just as easily:

```
(13) (exists y
       (lambda x
         (durative-event-has-interval
          meeting1 x))
       (exists z
        (lambda x
          (durative-event-has-interval
           meeting2 x))
        (overlap y z)))
```

This method of supporting metaphorical extension by explicitly defining the space of possible ways of conceptualizing an object allows us considerable flexibility in understanding novel metaphoric use.⁸

The same augmented coercion rules can be used if we wish to license a shadow relation that has no backend access code associated with it. We might want to use that strategy in the situation where the metaphoric use can be anticipated but the access code associated with the shadow would have to perform exactly the same computation as the coercion code.

5 Comparison with other accounts

As in DeJong and Waltz's work (1983), the King Kong coercion mechanism is triggered by violations of sort restrictions on arguments. We do not, however, agree with DeJong and Waltz's contention that "Nouns are far less likely to be metaphorical than verbs." The symbiosis between shadows and coercion rules implies that the metaphor lies not in the functor or its arguments, but rather in the association between them. Furthermore, our mechanism also structures the path between metaphoric use and ideal meaning, and provides computational support for argument coercion. The mechanism has the same advantage over the work of Jacobs and Martin.

5.1 Jacobs and Martin

In a series of papers (Besemer and Jacobs, 1987; Jacobs, 1986; Jacobs, 1987), Paul Jacobs has developed a relationship he calls a *view*. Views express a relationship between event types that implements metaphorical extension. For example, in order to handle examples like "The command takes three arguments", he defines the following view:

```
(VIEW execute-operation
  causal-double-transfer
  (ROLE-PLAY input object-1)
  (ROLE-PLAY output object-2)
  (ROLE-PLAY user source-1)
  (ROLE-PLAY operation source-2))
```

⁸Note that shadows always embody disjointness between at least one of their arguments and those in the ideal meaning. Thus, no input relation can be simultaneously interpreted both as a subsumed relation and as a shadow.

In Jacobs' system, this *view* would incorporate the metaphorical mappings from the full range of expressions referring to exchange operations such as giving, buying, and selling. As a result, the mappings in this *view* may be used to understand expressions such as "This command gives you the file names", and so on.

Like the work of Martin (see below), Jacobs' approach has the potential for grouping families of relationships into situations, a capability King Kong does not yet have. Jacobs' *views* correspond roughly to our shadow relations.

However, the *view* mechanism provides no limitations on the correspondences between the objects in the *ROLE-PLAY* declarations, nor does there seem to be any capability for computing one argument class from another. As a result, it is difficult to see how Jacobs' account would intelligently restrict the range of novel language use the system will handle, or how it might be used to provide computational support for sort coercion in an interface.

Martin (1987a, 1987b), working with the same mechanism, takes steps toward addressing the first concern. His work involves learning new metaphoric uses in light of already recognized metaphors. So Martin's heuristics allow the system to learn what "getting out of Lisp" means if it knows what "getting into Lisp" means. His system knows about entering and exiting, enabling and disabling Lisp processes, and that there is a map between entering and enabling Lisp. Because entering and exiting are closely connected (they are related by the frame semantic relation *reversible-state-change*), Martin's system can build the metaphoric link from exiting to disabling Lisp. Techniques such as this one constrain the interpretation of novel language use, since the system can only generalize from the existing library of metaphoric uses. However, they provide no computational support for *evaluating* novel uses.

5.2 Gentner et al.

Gentner's structure-mapping techniques (Gentner, 1983; Gentner *et al.* 1987) are applicable mostly to explicit analogies such as "An electric battery is like a reservoir." Her approach, implemented by Falkenhainer and Forbus (1986), maps the structure of the source of the metaphor to the structure of the target by creating match hypotheses between relational representations of the base and target using a set of match construction rules. But the central example of a match construction

rule seems to require that the names of the predicates in the facts being matched be identical. Under this sort of construction rule, it is possible to derive a metaphoric mapping only if the names of the predicates have been set up to encode the metaphor ahead of time. Under this system, it is not possible to deduce new metaphors; in fact, one can only recognize them if the metaphoric link has been made but not recorded.

5.3 Boguraev and Pustejovsky

Boguraev and Pustejovsky (1990) argue that the normal conceptions of the structure of the lexicon are impoverished for two major reasons. First, a great number of distinctions beyond those usually made are necessary to capture the essential aspects of lexical semantics. Second, the common technique for representing ambiguity in the lexicon (enumeration) falls short because enumeration of word senses neither organizes the senses intelligently nor provides for creative use of words.

For instance, under the enumeration method, the following uses of “fast” require that at least these three senses be listed in the lexicon:

- fast(1):** able to move quickly (a fast car)
- fast(2):** able to perform some act quickly (a fast typist)
- fast(3):** taking little time (a fast oil change)

However, these three senses are not enough to account for the creative use of “fast” in a phrase such as “a fast highway”.

Pustejovsky’s solution to this problem (outlined also in (Pustejovsky, 1990)) is a “generative lexicon”, which organizes lexical items with respect to one or more of: (1) argument structure, (2) event structure, (3) qualia structure, and (4) lexical inheritance structure. These lexical structures are intended to address the different ways in which words are understood; the differing interpretations of “fast” shown above are taken to be a function of the differing qualia structures of “car”, “typist”, “oil change”, and “highway”.

While Pustejovsky’s proposal for a variety of lexical structures is far richer than anything currently implemented in King Kong, one problem with his account is that the links are links between lexical items and not between objects in a domain model. Simple cases of anaphoric reference demonstrate that in many cases the coercions

that he conceives of are properties not of lexical items but rather of the objects referred to:

- John bought a Porsche, and it’s fast.
- John hired a typist, and he’s fast.
- I drove down I-90 yesterday, and it’s fast.
- John bought a new car, but Bill’s is faster.
- John hired a good typist, but Bill’s is faster.
- America is supposed to have good highways, but Italy’s are faster.

The lexical items whose qualia structures are intended to account for the different interpretations of “fast” are not present in the second clause of each of the preceding examples, but the correct interpretations are still available. This implies that it is the language user’s conception of the *object* in question (that is, the user’s world model) that determines the precise sense of “fast”.

In our account, in contrast, the links that support the range of metaphoric extensions Pustejovsky deals with reside in the domain model. This account also supports generalization of these extensions to hierarchies of semantic classes:

- John bought a new car, and it’s fast.
- John bought a new vehicle, and it’s fast.

and preserves these extensions under synonymy:

- John bought a new car, and it’s fast.
- John bought a new automobile, and it’s fast.

6 Conclusion

One insight missed in most relation-based accounts of metaphor⁹ is the wide space of possibilities for conceptualizing the argument types: how these possibilities are constrained, how the transformations can be computed. The coercion mechanism in King Kong supports metaphoric processes both statically and dynamically, by defining how metaphoric links between relations are established and supporting computational tools for comprehending and processing novel metaphoric uses.

Acknowledgments

This research was supported by the MITRE Corporation under MSR project 91340.

⁹With the exception of Boguraev and Pustejovsky’s, of course.

References

- [Besemer and Jacobs 1987]
David J. Besemer and Paul S. Jacobs. FLUSH: A flexible lexicon design. In *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics*, pages 186–192.
- [Boguraev and Pustejovsky 1990]
Branimir Boguraev and James Pustejovsky. Lexical ambiguity and the role of knowledge representation in lexicon design. In *COLING-90: Proceedings of the 13th International Conference on Computational Linguistics*, volume 2, pages 36–41.
- [Brachman and Schmolze 1985]
R.J. Brachman and J.G. Schmolze. An overview of the KL-ONE knowledge representation system. *Cognitive Science*, 9(2):171–216.
- [DeJong and Waltz 1983]
Gerald F. DeJong and David L. Waltz. Understanding novel language. *Computers and Mathematics with Applications*, 9(1):131–147.
- [Falkenhainer et al. 1986]
B. Falkenhainer, K.D. Forbus, and D. Gentner. The structure-mapping engine. In *AAAI-86: Proceedings of the Fifth National Conference on Artificial Intelligence*, pages 272–277.
- [Gentner 1983]
Dedre Gentner. Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, 7:155–170.
- [Gentner et al. 1987]
Dedre Gentner, Brian Falkenhainer, and Janice Skorstad. Metaphor: the good, the bad and the ugly. In Yorick Wilks, editor, *TINLAP-3: Theoretical Issues in Natural Language Processing-3*, pages 155–159, New Mexico State University, Las Cruces.
- [Herskovits 1986]
Annette Herskovits. *Language and spatial cognition: an interdisciplinary study of the prepositions in English*. Cambridge University Press, New York.
- [Herskovits 1989]
Annette Herskovits. The linguistic expression of spatial knowledge. L.A.U.D. Paper A 248, Linguistic Agency University of Duisburg.
- [Jackendoff 1983]
Ray Jackendoff. *Semantics and Cognition*. MIT Press, Cambridge, MA.
- [Jacobs 1986]
Paul S. Jacobs. Language analysis in not-so-limited domains. In *Proceedings of the IEEE Fall Joint Computer Conference*.
- [Jacobs 1987]
Paul S. Jacobs. A knowledge framework for natural language analysis. In *IJCAI-87: Proceedings of the 10th International Joint Conference on Artificial Intelligence*, pages 675–678.
- [Lakoff 1987]
George Lakoff. *Women, Fire, and Dangerous Things*. University of Chicago Press, Chicago.
- [Martin 1987a]
James H. Martin. The acquisition of polysemy. In *Proceedings of the Fourth International Workshop on Machine Learning*, pages 198–204.
- [Martin 1987b]
James H. Martin. Understanding new metaphors. In *IJCAI-87: Proceedings of the 10th International Conference on Artificial Intelligence*, pages 137–139.
- [Minsky 1975]
Marvin Minsky. A framework for representing knowledge. In Patrick Henry Winston, editor, *The Psychology of Computer Vision*, chapter 6, pages 211–277. McGraw-Hill, New York.
- [Pustejovsky 1990]
James Pustejovsky. Lexical ambiguity and the role of inheritance. Talk given at BBN, Cambridge, MA, 6 November 1990.
- [Talmy 1983]
Leonard Talmy. How language structures space. In Herbert Pick and Linda Acredolo, editors, *Spatial Orientation: Theory, Research, and Application*. Plenum Press, New York.