

# Preference Semantics, Ill-Formedness, and Metaphor

Dan Fass and Yorick Wilks

Cognitive Studies Centre  
University of Essex  
Wivenhoe Park  
Wivenhoe Park, Colchester  
Essex CO4 3SQ England

This paper is about the relationships between Preference Semantics (PS) and ill-formedness, and between Preference Semantics and metaphor. Two types of “preference”, declarative and procedural, are distinguished. The PS framework is examined with respect to notions of well- and ill-formedness, and two criteria for ill-formedness are distinguished, both of which are possessed by PS: an absolute criterion that corresponds to conventional notions of well- and ill-formedness, and a relative criterion that does not.

Four possible strategies are described for representing ill-formed input in general, and metaphors in particular. The strategies and the semantic representations produced by them are compared regarding their correspondence to human understanding (admittedly superficial given the shallowness of the PS representation) and their ability to produce correct sentence translations. We conclude that, because of the ambiguity of many individual and extended metaphors, two broad types of metaphor representation strategy are needed. A control mechanism is described that uses both these major types of strategy and that permits the temporary semantic representation of metaphorical ambiguity.

## 0. Introduction

We use the term “Preference Semantics” (PS) to indicate not programs that have parsed English into a semantic representation, nor the details of that semantic representation (all of which could have been different), but rather the underlying principles. The main principles or claims are as follows (and underlie the sequence of papers Wilks 1968, 1973, 1975, 1978). The last two will be of most concern to us here:

- a) It is possible to pass from English to a semantic representation without a module devoted explicitly to syntactic analysis, and without traditional syntactic classification of words or sentence components (for example, N, NP, VP). The necessary generalisations for parsing can all be expressed in the terms needed for the semantic representation. Moreover, these need not result in any kind of text “skimming” that misses essential features of the text and its content.
- b) The representation need not be of the model theoretic type, and the classic problems of quantification, etc., can be dealt with by special proce-

- dures.
- c) The description of the representation and the procedures that generate it should all be procedural and, most important, the representation should be the product of a few, general, and autonomous (not content-dependent) procedures. Moreover, the procedures should be consistent with a Least Effort principle of language understanding (Wilks 1975).
- d) The representation is based on a set of semantic primitives, of differing types (actions, substantives, qualities, etc.), but no claims are made that the set is universal: there could be many alternative sets for special tasks, domains, or cultures. All that is required is there be *some* privileged set to generate a representation.
- e) The representation emphasises the linear, rather than the recursive, properties of language: its structure therefore emphasizes linear boundaries of clauses and phrases (but with no special role for sentences) as a basis for a surface representation from which progressively deeper representations can be obtained by inference. The repre-

Copyright 1984 by the Association for Computational Linguistics. Permission to copy without fee all or part of this material is granted provided that the copies are not made for direct commercial advantage and the *Journal* reference and this copyright notice are included on the first page. To copy otherwise, or to republish, requires a fee and/or specific permission.

0362-613X/83/030178-10\$03.00

sentational item corresponding to the piece of language between two such boundaries (whether a word or a sentence) we call a *template*, which is a complex structure (see below) having no associations with the term as used to denote a string of surface items, as in vision analysis.

- f) The representation is formed upon a principle of preference for the “best fit”: thus, there is no single correct representation (except in special circumstances) for a text string, but the best, most internally coherent representation, *chosen from among competing representations*. Representational structures can be seen as “preferring” other associated representations, and an overall representation for a text is produced by allowing maximal satisfaction of such preferences, which will mean (as in the political analogy on which the notion is based) that some constituent representations do *not* have their local preferences satisfied.
- g) The last representational principle has a correlate at the level of text relationships: ill-formedness (and, we shall claim below, metaphor) is not a binary, yes-no, matter but a function of representational satisfaction, which includes being a function of the state of the dictionary for the words and higher level items constituting the text. To put it crudely, ill-formedness is a matter of what the analysis system believes the dictionary and state of the world to be, and how far it can be *extended* by rule with the aid of the knowledge structures available. To use an example from Wilks (1978)

(1) The car drank gasoline

will be ill-formed or not, depending on what you believe about drinking and about cars (thus crossing what would be, for many, a semantic-pragmatic boundary), and similarly for

(2) John ran a mile

depending on beliefs about running and distance (and so similarly for the so-called syntax-semantics distinction and the class of “intransitive verbs”).

It is part of principle (a) above that preference is a syntactic notion as well as a semantic notion in that one general rule can deal with both sorts of conventionally distinguished phenomena. Thus (2) is ill-formed just because [run] prefers no object, just as [believe] prefers a propositional object (a full *template* in the terms set out below) but will accept a human object nevertheless. However, in this short paper we arbitrarily restrict ourselves to phenomena that would conventionally be considered matters of word-sense semantics.

On this view, much of what has often been thought of as ill-formed – particularly violations of Katzian selection restrictions (Katz and Postal 1964) – is not only not ill-formed but is typical of normal usage, and must not be rejected if it can be accommodated by the procedures of PS. The emphasis here is rather different from the standard one: on the PS view, the violation of preferences (such as those of *drink* for an animate agent or a liquid object) is the norm, and must not be treated as an exceptional matter, outside the core of English. Such locutions are statistically so normal and understood even when wholly novel, that their representation and processing must be performed as part of the *central processes* of a language understander.

Some of the above points can be found incorporated in other language understanding systems, for example Schank and his associates (Schank 1975) for (a) – except for their predilection for NPs – (b), (d) – except for their insistence on a universal set of primitives – and more recently (e). For (b) almost any classical example semantic net system (Simmons 1973, Hendrix 1975). What we shall do here is develop the last two principles towards a general theory of the understanding of ill-formed and metaphorical language.

The concrete setting of our current research is the construction of a semantics/knowledge-based spelling corrector, but we shall not emphasise that here.

### 1. A Brief Resume of the Preference Semantics System

The following terminology will be useful: a *semantic formula* is a representation of a word-sense; it contains a *head*, which represents the “main element” in the sense – for example, whether a noun refers to a MAN or a THING, or whether a verb denotes an act of THINKing, or of DOing. Its internal structure is of left-right dependency.

The following is a simplified semantic formula for the action *drink*:

- (3) ((\*ANI SUBJ) (((FLOW STUFF) OBJE)  
(MOVE CAUSE)))

Reading the formula for *drink*, it is an action, preferably done by animate things (\*ANI SUBJ) to liquids ((FLOW STUFF) OBJE). The SUBJ case displays the *preferred* agents of actions, and the OBJE case the *preferred* objects, or patients.

A *template* is a structure, based on slots for three semantic formulas that can themselves have dependent formulas, such that the whole structure represents a possible “message”. A template can have any number of formulas (from one to any). Each fragment of a sentence (clause or phrase) has templates matched onto it during parsing and the existence of more than one template per fragment is representational ambiguity, to be reduced by examining the internal “fit” of

templates, and the external relations between templates for successive fragments of text.

The formulas in each template are determined to see if their preferences are satisfied. In what follows, [square brackets] denote the formula for a word. So, for example, [crook (man)] denotes the formula for the human sense of the word *crook*.

The sentence

(4) The policeman interrogated the crook

will produce two candidate interpretations, which are templates of formulas, written left to right, filling its action-agent-object slots

[policeman] [interrogated] [crook (man)]

[policeman] [interrogated] [crook (thing)].

So, we have two possible template representations (that is, two possible readings) for the sentence.

The *template expansion algorithm* seeks to resolve this: it looks into subparts of the formulas to see if any preferences are satisfied. [interrogate] prefers a human actor; this is marked in both representations. It also prefers a human object: [crook (man)] can satisfy this preference, but [crook (thing)] cannot.

So we have (in the following, → or ← represents satisfied preferences)

(4a) [policeman] → [interrogates] ← [crook (man)]

(4b) [policeman] → [interrogates] [crook (thing)]

The first of these has the larger number of satisfied preferences, or greater “semantic density”, so it is preferred. The template representation chosen here, the one with the highest semantic density, has full preferential links between every pair of formulas.

In the case of a sentence like (1) that contains a failed preference (whether or not it is metaphor, for example *The VDU interrogated the crook*), the first reading is accepted because there are no other competing readings.

## 2. Three Types of Dictionary Information

The semantic information in dictionary entries (formulas) can be categorised into three types, which will be exemplified in the semantic formula for *drink* (3).

(i) *Inherent information*: “data”

The semantic properties that a dictionary entry contains *specifically about the item itself*. In a semantic formula, the main example of this is its head primitive(s), for example (MOVE CAUSE).

(ii) *Label information*: “labels”

Case information describing the case relationships between a dictionary entry and other dictionary entries. Label information exists in the case subparts of semantic formulas

as case primitives like SUBJ (to be interpreted as AGENT) in (\*ANI SUBJ), and OBJE in ((FLOW STUFF) OBJE).

(iii) *Contextual information*: “expectations”

The inherent semantic information that a dictionary entry expects *other dictionary entries* to possess as inherent information. Like label information, contextual information exists in the case subparts of semantic formulas as semantic primitives or subformulas like \*ANI and (FLOW STUFF).

When disambiguating word-senses, all three types of information are used. In section 1 above, we saw how the template expansion algorithm resolved (4): [interrogate] prefers a human object, where “object” is label information, and “human” is contextual information. [crook (man)] satisfies this preference because its head primitive – inherent information – is human.

We wish to distinguish dictionary entries that contain semantic contextual information and those that do not:

► *predicates*

Contextual information occurs in the semantic formulas for verbs, adjectives, nominalised verbs, and idioms (Wilks 1975, Boguraev 1979). Dictionary entries for prepositions, called paraplates (Wilks 1975) or preplates (Boguraev 1979), larger structures that tie templates together and have the function of inference rules, also contain contextual information because they specify the semantic class of head noun or verb being modified and the head noun of modifying prepositional phrase, but they are outside the scope of discussion here.

► *non-predicates*

Simple nouns like *table*, *car*, and *chopper*, which do not contain contextual information in their semantic formulas at the top level (that is, [car] might contain coding that humans use cars to achieve a goal, but that would not appear at the top level of the “goals of cars”).

By “predicate” we mean *specifically* dictionary entries containing semantic contextual information at the top level, and *not* the more general use of the term.

## 3. Two Types of “Preference”

This section examines the notion of preference and makes an important distinction between a declarative and a procedural version of preference (Fass 1983).

### 3.1. Preference-as-restriction

A preference is (dictionary) information in a semantic formula expressing some kind of restriction on the semantic context in which a word-sense can occur.

Two observations:

*Preferences-as-restrictions are binary.*

A preference is either satisfied or violated: it cannot be partially satisfied. This is because of the organisation and generality of PS semantic primitives, which are hierarchically organised but only at two levels of generality. For example, the “class element” primitive \*ANI includes the class of primitives (BEAST, MAN, FOLK, SIGN, or THIS), that is, any animate entity. There can be no partially satisfied preferences with the present set of primitives, as *would be* the case if BEAST could satisfy a preference for MAN because both are in the class \*ANI.

*A preference is a piece of contextual information.*

Although a preference coding occurs within a case subpart of a formula, the corresponding label information is *not* part of that preference.

As preferences-as-restrictions are contextual, it is only predicates that have them in PS. But if preferences-as-restrictions referred instead to inherent information, then non-predicates would also have preferences. Consider the helicopter meaning of the word *chopper*, whose formula has the head primitive THING (that is, physical object). If a preference described inherent information, then we could view choppers as *preferring* to be THINGS but not having to be THINGS. We shall consider just this in section 6.

### 3.2. Preference-as-procedure

Preference is viewed as a procedure for assigning scores to competing alternative representations and choosing the best one. In PS, preference-as-procedure uses as its criterion for choosing between competing sentence readings the number of preferences-as-restrictions that are satisfied.

The four key elements of preference-as-procedure are:

- ▶ *production* – it produces all sentence readings whether or not they contain preference violations;
- ▶ *scoring* – readings are scored according to how many preference satisfactions they contain;
- ▶ *comparison* – whether or not an individual reading is accepted depends on a comparison with other readings;
- ▶ *selection* – the best reading (that is, the one with the most preference satisfactions) is taken, even if it contains preference violations.

By choosing the best available, preference-as-procedure as a single procedure has *two effects* when it operates: it disambiguates word-senses and at the same time provides system robustness (that is, a sentence reading is always returned).

It should be emphasised that preference-as-procedure is a general strategy, used to provide disambiguation and robustness at many different levels in

the PS system, not just with preferences-as-restrictions. The two types of preference are separable from each other: preferences-as-restrictions can be used by other procedures, and preference-as-procedure can be used with other types of dictionary information.

## 4. The Preference Semantics System and Ill-Formed Input

### 4.1. Preference Semantics and ill-formedness

We can best understand a Preference Semantics approach to ill-formedness by comparing it with Katz and Postal’s (1964) semantic markers/selection restriction approach. Katz and Postal’s approach embodies a binary principle of semantic well-formedness similar to that assumed in standard generative syntax: well-formed and ill-formed.

A selection restriction is binary – a semantic marker either fits a selection restriction or it does not. Preferences-as-restrictions, as they appear in semantic formulas, are also binary (and equivalent to selection restrictions): a semantic class either satisfies a preference or it does not. With the binary principle, there is an *absolute* criterion for ill-formedness: a semantic relation can be labelled ill-formed by examining *that relation alone*, without looking at any others.

At the level of the constituent or sentence, preference-as-procedure is different from a selection restrictions approach. This should be clear if we examine a selection restrictions approach using the same four elements we used for preference-as-procedure:

- ▶ *production* – only those sentence readings with all their selection restrictions fulfilled are produced;
- ▶ *scoring* – there are only two scores – (i) “well-formed”: all selection restrictions fulfilled, or (ii) “ill-formed”: one or more restrictions are violated;
- ▶ *comparison* – none. Readings are considered individually, without comparison against other readings;
- ▶ *selection* – the sentence reading with all selection restrictions fulfilled is taken, if such exists.

The preference approach adopts a different, unary principle of “formedness”. If a preference in a sentence is violated, then *a reading is still produced* for that sentence, so being “formed” is like being well-formed in the selection restrictions sense.

But whether that (preference violating) reading is accepted as if it was well-formed, or rejected as if it was ill-formed, depends on whether there are other possible readings for that sentence and on the nature of these readings:

- ▶ The reading is *accepted* if either there are no other readings for the sentence or if all the other readings for the sentence have more prefer-

ence violations. In such situations, the PS system assumes that the writer meant to produce the reading, that is, that it is some novel use of language (for example, metaphor) and is well-formed.

- The reading is *rejected* if there is another reading for the sentence that has fewer preference violations. However, being rejected in this way is probably not tantamount to being ill-formed because, in some other circumstances, sentences containing a preference violation (like the rejected reading) could be accepted as the best available.

If all the preferences are fulfilled in a reading of a constituent, then, although the constituent/sentence may be “well-formed” in the selection restrictions sense, that reading may not necessarily be accepted. This is because there may be another reading of the same sentence that also has all of its preferences satisfied and is equally acceptable.

So, the difference between PS and Katz and Postal’s approach is at the procedural level. With the unary principle of PS, the criterion for ill-formedness is *relative*: a reading can only be labelled “ill-formed” *after comparing it with other readings*, and not by examining that reading alone, which is why preference-as-procedure produces *all* readings, whether or not they contain preference violations.

So, we have distinguished two criteria for ill-formedness: absolute and relative. Within PS, the criterion of absolute ill-formedness is used for the semantic relations between individual word-senses (3.1.), and relative ill-formedness for readings of constituents of sentences (3.2.).

#### 4.2. The nature of preference violations

Preference violations between two words can be caused either by some “total” mismatch of word-senses, as between [interrogates] and [crook (thing)] in (4b); or by some metaphorical relation, as there is between [car] and [drink] in (1) *The car drank gasoline*. Examining the preference violation itself does not reveal its nature; we can only discover the type of preference violation by examining competing readings (if any), which is what preference-as-procedure does. If all the other readings have more preference violations, then the reading containing the single preference violation is assumed to be appropriate and a metaphor.

However, we can produce sentences containing a metaphor in which examining the alternative sentence readings cannot help establish what type of preference violation we have. Consider the sentence

(5) That chopper drinks gasoline

which contains a metaphor (Van Eynde 1982).

There are two readings of the sentence, based on the ambiguity of *chopper* as either “ax” or “helicopter”. The two template representations are:

(5a) [chopper (helicopter)] [drinks] ← [gasoline]

(5a) [chopper (ax)] [drinks] ← [gasoline]

Both [chopper (helicopter)] and [chopper (ax)] have the semantic head THING (physical object), and both violate the preference of [drink] for an ANIMATE agent. In this example, the PS system cannot discriminate between the two sentence readings – one containing mismatched word-senses (5b), the other containing a metaphor (5a) – in terms of their number of satisfied preferences. So it is unable to decide which reading is metaphorical (and appropriate).

Because a preference violation locates failed semantic relations, we can try to determine whether or not that violation is caused by a metaphor by applying additional semantic information there. In the next section we consider the sort of semantic information necessary to resolve (5) and one suggested way of representing that information.

#### 5. Semantic Information about Metaphor

Van Eynde (1982) has pointed out that the standard PS system cannot choose the correct reading from templates (5a) and (5b) above. He suggested a set of rules, *polysemy rules*, that can recognise one of the violations as being caused by a metaphor and choose the correct reading.

Polysemy rules are applicable to metaphors involving a predicate and a non-predicate; they can be used not only to choose between readings like (5a) and (5b) but also to confirm that a single reading produced for a sentence like (1) is a metaphorical one. Metaphors between two non-predicates, for example *This encyclopaedia is a gold-mine* (Rumelhart 1979), are excluded from consideration in this paper.

It is very important to divorce two issues concerning PS and metaphor: first, ways of *recognising* and choosing a reading containing a metaphor, that is, polysemy rules, described in section 5.1. below; second, possible strategies for *representing* that metaphorical reading, described in section 6. Polysemy rules can be combined with a number of those strategies.

##### 5.1. Polysemy rules

What is essential first of all is to provide additional semantic information to distinguish the vehicle sense from the ax sense of *chopper*. Van Eynde introduces a new primitive VEHICLE, which he uses as head primitive of the vehicle sense of *chopper*.

A polysemy rule looks like this:

- (6) condition: *certain environmental data*, such as:  
 A is the AGENT slot of a template  
 and B is an action in the ACTION

slot of the same template. Subject preference of B = ANIMATE. Head primitive of A = VEHICLE.

assignment: Head primitive of A := ANIMATE.

The format of the above we take to be self-evident. The rule would normally be understood to run its assignment whenever the condition is satisfied. On a historical note one can compare polysemy rules with the very general dictionary extension rules of Givon (1967).

The effect of this particular rule is to *change data*, that is to alter the head primitive of the helicopter sense of *chopper*. Note that, with rules of this type, the assignment can either

- ▶ change the data by modifying the inherent semantic information in the non-predicate (thus making it animate), so that the unchanged semantic formula for *drink* (preferring an animate agent) will still pick out this reading; or,
- ▶ alternatively, one could change the expectations, modifying the semantic formula for *drink* (the predicate), so that it accepts vehicular agents as second best to genuinely animate ones; or,
- ▶ one could modify [drink] more radically, by changing *its* inherent data (see below); or,
- ▶ we could just leave both formulas unchanged.

We will consider these four alternatives in section 6.

## 5.2. Discussion

The first point to note is that polysemy rules alone do not provide a means of recognising the initial conflict between *chopper* and *drinks*, and does not provide a means of selecting the sentence reading containing the correct sense of *chopper*. Thus, polysemy rules cannot operate on their own but only within some more general word-sense disambiguation mechanism such as PS, in some such way as the following: for sentence (5), only after the template expansion algorithm of PS has produced the two readings (5a) and (5b) can polysemy rules be applied to the non-predicate involved in the preference violation, and the template expansion algorithm tried again. One of the readings for the sentence will now have no preference violations

(5c) [chopper (helicopter)] → [drinks] ← [gasoline]

and is accepted.

In the foregoing (5.1.), we have embedded Van Eynde's polysemy rule (6) within some general PS environment for making choices between readings after (6) has altered the available readings. It was necessary to do this because, as we pointed out, the rule alone does not specify how to select readings. Moreover, Van Eynde sees rules like (6) as operating within a production system. If that production system was uncontrolled, then such rules would run whenever their conditions were satisfied. The control regime for

those rules is hard to imagine, and would certainly be very complex.

## 6. The Representation of Metaphor and Ill-Formed Input

In this section we describe and compare four strategies for representing ill-formed input in general and metaphors in particular, in semantic representations. It is assumed that a process with the power of that described in section 5 above has located a preference violation or "semantic conflict" and recognised it as being a metaphor.

### 6.1. Four strategies for the representation of metaphor

We will illustrate these strategies using sentence

(1) The car drank gasoline

though we could also have considered reading (5a) of sentence (5) as an example. The best reading for (1) has a conflict between the expectation of the predicate [drink] expecting an animate agent as subject and the data in the non-predicate because the actual subject (the car) is inanimate. If we built a semantic representation of this sentence, then the conflict would remain in the representation.

Obvious strategic choices are:

- (i) *Passive* strategy  
Relax the preference of the predicate and accept the semantic representation with the conflict unresolved (Wilks 1975); at no point are data or expectations changed, and the analysis system simply accepts the representation it is given.
- (ii) *CTD*, or Change The Data, strategy  
Change the inherent data in the non-predicate in such a way that it meets the expectations (Van Eynde 1982). So, in sentence (1) alter the data and replace the head primitive VEHICLE in [car] by the primitive ANIMATE in the semantic representation. This is one top-down (expectation driven) approach: in the case of conflict between what you have and what you expect, change what you have and be guided by your expectations.
- (iii) *CTE*, or Change The Expectations, strategy  
Change the expectations in the predicate in such a way that they meet the data (Van Eynde 1982). So, for sentence (1) alter its semantic representation by changing the expectation that the subject of [drink] must be ANIMATE to VEHICLE
- (iv) *Active* strategy  
A more radical approach, explored in Wilks (1978), would produce a completely new formula [drink] by rule and equivalent to [consume], modifying inherent and expectational data, so as to accept an animate agent (car). This approach uses the wider context of frame-like representations, called pseudo-texts, in addition to semantic

formulas. At its crudest the method consisted of finding particular facts (when faced with (1)) about cars in its frame-like data base such that cars did operate on gasoline in a manner semantically related to drinking. The only fact located was "cars consume gasoline" and so a [drink] had a new representation added, namely the appropriate formula from the dictionary entry for [consume]. This is a top-down, knowledge-driven approach, but cannot be termed CTE or CTD since no formula of *drink* is modified but a new one slotted into the templates for *that particular ill-formed locution*. We shall compare this method with the others above, that need less detailed and cumbersome context than frame methods and are more narrowly semantic.

## 6.2. Comparison of the four strategies

The strategies are compared in two ways. First, the degree to which the semantic representations containing metaphors produced by the different strategies correspond to human understanding of those metaphors. Given the shallowness of a PS representation, that correspondence can be no more than superficial. Secondly, whether or not the semantic representations of the different strategies would assist in concrete computational tasks, such as producing correct translations.

Most, if not all, individual metaphors can be read or understood in two ways. For example, the metaphor in (1) can be understood either by viewing the predicate *drink* as the car-like consuming of petrol, or by seeing the non-predicate *car* as having some human properties. Within PS, the CTE strategy and the active strategy reflect the first, predicate reading by altering semantic information in the predicate; the CTD strategy reflects the second, non-predicate reading by changing inherent information in the non-predicate. No single strategy reflects *both* readings. By leaving the preference violation in the semantic representation, the passive strategy does not reflect *either* reading and does not reflect human understanding of metaphor at all.

In extended metaphors (those beyond a single clause), the initial metaphorical reading can be carried over in either the non-predicate or the predicate. Consider the following extended metaphors that are also cases of gapping (Hankamer 1973):

- (7) The car drank gasoline and (the car) purred to itself
- (8) The car drank gasoline and the taxi (drank) diesel

In (7), the metaphorical usage of the non-predicate *car* is continued; in (8), it is the predicate *drink*.

We now examine how closely the strategies of 6.1. reflect our understanding of extended metaphors like (7) and (8). To do this, we shall assume a simplified form of rules for filling dummy template nodes (Wilks 1975). Those more familiar with Chomsky (1977) can think of this in terms of a form of trace mechanism in which the trace node in the template representing the second clause inherits information from the controlling node in the first clause. Hence in (7) the formula of *car* will be inherited by the empty agent node in the template containing [purr].

Let us consider (7) first. What happens when each strategy encounters [car] and [drink] in the first clause of the sentence, and then encounters [car] inherited from the first template and [purr] in the second clause?

When the CTD strategy encounters [car] and [drink], it removes the preference violation between them by reassigning VEHICLE as ANIMATE in the non-predicate [car]. This modified formula of [car] is inherited from the first template; [purr] expects an animate SUBJ and [car] is now ANIMATE, so there is no preference violation between them.

The CTE strategy removes the preference violation between [car] and [drink] by changing the SUBJ preference of the predicate [drink] from ANIMATE to VEHICLE. [car] is unchanged and is inherited unchanged. Because [car] is still marked as inanimate, there is a preference violation with *purr*, which causes the CTE strategy to alter the SUBJ preference of [purr] to VEHICLE.

The passive strategy does not change either [car] or [drink], leaving the preference violation between them. A second preference violation is left in the second clause as well.

With the active strategy, a car-frame (or pseudo-text) is used, and [drink] would have a new *consume* sense and there would be no effect on [car]. Hence, the frame would be accessed again for the second clause, but would either find no new sense for *purr* in the limited context of *to itself* (which would become just a passively accepted, though preference-violating, template) or it could hope to re-apply the active strategy and find from the car frame that the only noise cars were noted as making (other than in conditions of trouble where they would *backfire*, etc.) was *hum*, which could be imposed in place of [purr], and would be confirmed by a causal inference from the beneficial effect of [consume gasoline]. However, this might be difficult to embody in a serious knowledge representation since there is no non-metaphorical description in English of the noise of cars.

So for (7) the active and passive strategies both leave preference violations in the second clause. The CTE and CTD strategies do not, but of these two, the CTD strategy more closely reflects human understanding.

Now let us examine (8), *The car drank gasoline and the taxi diesel*. When processing (8), the CTD strategy changes semantic information in the non-predicate [car]. [drink] is unchanged and is inherited unaltered by the second template. [taxi] is inanimate, but [drink] expects an animate subject, so there is a preference violation, which will cause the semantic information in [taxi] to be changed in its turn.

The CTE strategy will change the SUBJ preference of the predicate [drink] to VEHICLE. This modified version of *drink* is then inherited by the second template. As [taxi] is a VEHICLE, there is no preference violation between [drink] and [taxi].

The passive strategy changes neither set of information, which leads to preference violations in both clauses. The active strategy would construct a new *consume* sense for [drink] that would be inherited by the action node of the second template. As [taxi] is a VEHICLE, there would be no preference violation between [taxi] and the new sense of *drink*.

In (8), where the metaphorical usage continued in the predicate, the CTE and active strategies most closely reflect human understanding because both have the effect of changing the predicate's expectations of its subject. However, in (7), where the metaphorical usage continued in the non-predicate, the CTD strategy was best because it changed the inherent data in the non-predicate.

If we take the production of correct translation as a *minimum* constraint on interpretation strategy, then the changes the four strategies make to semantic representations are important because the effect of one strategy can be to produce a correct translation while another can cause a mistranslation.

Consider

- (9) The car drinks gasoline and (the car) does not work well

where the metaphor in the first clause does not extend to the gapped second clause. Assuming a node inheritance mechanism once again, [car] will be inherited in the second clause.

If the non-predicate [car] is inherited unaltered, then that sentence is translated correctly as *La voiture boit de l'essence et ne march pas bien* because *marcher*, the appropriate translation of *work*, expects an inanimate subject. It is because they leave [car] unchanged that the passive, CTE, and active strategies all produce the correct translation of (9).

However, the CTD strategy reassigns [car] as ANIMATE, and this modified formula of *car* is inherited into the second template. The effect of this is to translate the sentence wrongly as *La voiture boit de l'essence et ne travail pas bien* because *travailler*, another translation of *work*, expects an animate subject.

(9) is not meant to be taken as decisive evidence in favour of the CTE strategy or the frame-based active

strategy. We are sure that sentences can be found where altering the predicate's semantic information would cause mistranslations, where only the CTD or passive strategy would produce correct translations (there are probably sentences for which the passive strategy would produce mistranslations too): a strategy that produces a correct translation for one sentence may well mistranslate another. It is not possible to pursue these possibilities in detail here because it would involve too much detail of the mechanisms by which a translation equivalent in the target language is located – for example, by a full semantic matching as in the MARGIE system (Schank et al. 1973), or from a prior guidance to possible target equivalents, as in Wilks (1973). That degree of detail would change the emphasis of this paper, in which translation is no more than a minimum condition that semantic strategies dealing with ill-formedness must meet.

Because individual metaphors are ambiguous, that is, can be read or understood in two directions, no one strategy is adequate. The passive strategy is totally unsatisfactory. Strategies that alter the semantic information of non-predicates (CTD strategy) are inappropriate for predicate readings of individual metaphors and for extended metaphors that continue a predicate reading such as the one in sentence (8). Equally, we cannot have only strategies that alter the semantic information of predicates (CTE or active strategy) because of both non-predicate readings of individual metaphors and extended metaphors continuing a non-predicate reading like (7).

As a result of the preceding comparison of strategies in terms of correspondence to human understanding and production of correct translations, it is clear that *both* strategies that change expectations and strategies that change data are needed. Since both these major types of strategy are fallible, how will the proper strategy be selected?

In the next section we propose a control mechanism using both types of strategy that makes the correct selections (in terms of human understanding and accurate translations above), that is, it allows individual metaphors like the one in (9) to be represented by both types of strategy, selects the CTE strategy for examples such as (8), the CTD strategy for those such as (7), and no strategy at all for sentences like

- (10) The cat drank milk and the dog (drank) water that do not contain metaphor.

### 6.3. Control of the strategies

In this section we consider only single representative examples of a strategy that changes expectations and one that changes data: these will be the CTE and CTD strategies. We limit our demonstration of the control mechanism to the sentences of 6.2. containing a gap-



ped clause – that is, (7), (8), (9), and (10) – though we believe it to be generally applicable.

We shall deal with the case of no metaphor first. If no metaphor is found in the first clause, as in (1), then a single template with the largest number of preferences is chosen in the normal way (see section 1).

If, as in (7), (8), and (9), a metaphor is encountered in the first clause, then both major types of strategy are applied, producing *two* competing templates for the clause representing metaphorical ambiguity, that is, the two possible readings of the metaphor (data and expectations important to the metaphor are included below):

(11a) [car (VEHICLE)] → [drinks (SUBJ VEHICLE)]\* ← [gasoline]

(11b) [car (ANIMATE)]\* → [drinks (SUBJ ANIMATE)] ← [gasoline]

Any semantic formula whose semantic information has been altered is marked by the control mechanism (indicated above by an \*). The template (11a) produced by the CTE strategy has an altered predicate [drink]; the template (11b) produced by the CTD strategy has an altered non-predicate [car].

If the second clause is a case of gapping, then the dummy node in the second template is analysed. If there is a single (unmarked) template representing the first clause, then the first clause did not contain a metaphor and the dummy node in the second template inherits the semantic formula from the controlling node in the first template in the way described earlier (section 6.2.). Hence, for (10), [drink] is inherited.

If there are two (marked) templates representing the first clause, as with (11a) and (11b), then a metaphor is present. Though the mechanism also operates if the dummy node in the second template is a predicate (as in (8)), let us suppose that the missing node is a non-predicate, as in (9) *The car drinks gasoline and does not work well* or (7).

To allow for individual metaphors like (9), the control mechanism assumes that the metaphor in the first clause has not been continued in the second: an *unaltered* version of the non-predicate is placed in the dummy node of the second template, taken from the template with an altered *predicate* because it contains the unaltered non-predicate. So, for sentence (9), the unaltered [car (VEHICLE)] is taken from the template with the altered predicate (11a), and a new template for the second clause (shown below in much simplified form) is produced:

(12) [car (VEHICLE)] → [works (SUBJ VEHICLE)]

If there is no preference violation between that unaltered non-predicate and the other nodes of the second template, then, provided no other reading has more

satisfied preferences, it is *that* reading of the template that is accepted.

If, though, we have a case of extended metaphor as in (7) *The car drank gasoline and purred to itself*, then there is a preference violation between the unaltered non-predicate [car (VEHICLE)] and the predicate in the template for the second clause. So, for (7), the following template (much simplified) is produced:

(13) [car (VEHICLE)] [purred (SUBJ ANIMATE)]

(13) must have more satisfied preferences than any other competing template but – and here the control mechanism departs from the standard preference-as-procedure – even if (13) has more satisfied preferences than any other template, it is not accepted as it is, because it contains a preference violation between [car] and [purr]. Instead, a new template for the second clause is created: its empty node is filled with the *altered* version of the same formula [car (ANIMATE)], inherited from the other template representing the first clause (11b), the one containing the amended non-predicate:

(14) [car (ANIMATE)] → [purred (SUBJ ANIMATE)]

This template is accepted if it has more satisfied preferences than any other. Because the second case of inheritance was from the template containing the amended non-predicate, the control mechanism knows that the CTD strategy was appropriate for the first clause: the template containing the amended non-predicate appears in the semantic representation for the sentence as a whole. Hence the control mechanism handles cases of extended metaphor like (7) and (8).

However, for sentences containing a single metaphor such as (9) and (1), the ambiguity of the metaphor remains unresolved as two possible templates, (11a) and (11b). In terms of the means of comparison used in 6.2. (correspondence to human understanding and production of correct translations), there is no need to keep both templates, so the template with the altered predicate is retained (the product of the CTE or active strategy), somewhat arbitrarily, because we believe this reading to be the more common of the two.

### Acknowledgments

This research is currently supported by Science and Engineering Research Council contract GR/C/44938, “Intelligent knowledge-based spelling correction”, and by the European Community DGXIII, Luxembourg, under contract ETL-1-E, “Linguistics for machine translation”.

The authors would like to thank Doug Arnold and Claire Grover for their many helpful comments and suggestions, and one of our reviewers for his/her com-

ments about selecting between CTD and CTE strategies.

## References

- Boguraev, B.K. 1979 Automatic Resolution of Linguistic Ambiguities. Technical Report No. 11. Computer Science Department, University of Cambridge, England.
- Chomsky, N. 1977 On Wh-Movement. In Culicover, P.; Wasow, T.; and Akmajian, A., Eds., *Formal Syntax*. Academic Press, New York: 71-132.
- Erlandsen, J.; Van Eynde, F.; McNaught, J.; Somers, H.; and Destombes, L. 1982 Dictionary and Semantics in Eurotra. Eurotra Contract Report ET-10-SEM. European Communities, Luxembourg.
- Fass, D.C. 1983 Preference. In McNaught et al. (1983) Part III, Section I.
- Givon, T. 1967 Transformation of Ellipsis, Sense Development and Rules of Lexical Derivation. Memo SP-2896. Systems Development Corporation, Santa Monica, California.
- Hankamer, J. 1973 Unacceptable Ambiguity. *Linguistic Inquiry* 4: 17-68.
- Hendrix, G. 1975 Expanding the Utility of Semantic Networks Through Partitioning. *Proceedings of the Fourth International Joint Conference on Artificial Intelligence*. Thilisi, USSR: 115-121.
- Katz, J. and Postal, P. 1964 *An Integrated Theory of Linguistic Description*. MIT Press, Cambridge, Massachusetts.
- McNaught, J.; Arnold, D.; Bennett, P.; Fass, D.C.; Grover, C.; Huang, X.; Johnson, R.; Somers, H.; Whitelock, P.; and Wilks, Y.A. 1983 Structure, Strategies, and Taxonomy. Eurotra Contract Report ETL-1-E. European Communities, Luxembourg.
- Rumelhart, D.E. 1979 Some Problems with the Notion of Literal Meanings. In Ortony, A., Ed., *Metaphor and Thought*. Cambridge University Press, Cambridge, England: 78-90.
- Schank, R.C. 1975 *Conceptual Information Processing*. North Holland, Amsterdam, Holland.
- Schank, R.C.; Goldman, N.; Reiger, C.; and Riesbeck, C. 1973 MARGIE: Memory, Analysis, Response Generation, and Inference in English. *Proceedings of the Third International Joint Conference on Artificial Intelligence*. SRI, Menlo Park, California: 255-261.
- Simmons, R.F. 1973 Semantic Networks: Their Computation and Use for Understanding English Sentences. In Schank, R.C. and Colby, K.M., Eds., *Computer Models of Thought and Language*. W.H. Freeman, San Francisco, California: 63-113.
- Van Eynde, F. 1982 Ambiguity. In Erlandsen et al. (1982), Chapter 5.
- Wilks, Y.A. 1968 Computable Semantic Derivations. Memo SP-3017. Systems Development Corporation, Santa Monica, California.
- Wilks, Y.A. 1973 An Artificial Intelligence Approach to Machine Translation. In Schank, R.C. and Colby, K.M., Eds., *Computer Models of Thought and Language*. W.H. Freeman, San Francisco, California: 114-151.
- Wilks, Y.A. 1975 A Preferential Pattern-Seeking Semantics for Natural Language Inference. *Artificial Intelligence* 6: 53-74.
- Wilks, Y.A. 1978 Making Preference More Active. *Artificial Intelligence* 10: 1-11.