Computers Math. Applic. Vol. 23, No. 2–5, pp. 205–221, 1992
Printed in Great Britain. All rights reserved

0097-4943/92 $5.00 + 0.00

THE PREFERENCE SEMANTICS FAMILY

YORICK WILKS
Rio Grande Research Corridor, Computing Research Laboratory
New Mexico State University, Box 30001, Las Cruces, NM 88003, U.S.A.
DAN FASS
Centre for Systems Science, Simon Fraser University, Burnaby, BC V5A 1S6, Canada

Abstract—The paper discusses the origins and structure of Preference Semantics, a procedural and computational system for extracting the meaning structure of natural language texts, based on notions of "maximal semantic density" and coherence. The basic representational structures and procedures of Preference Semantics are described, as well as the forms these notions have taken in the work of others.

1. INTRODUCTION

Preference Semantics (PS from now on) has its philosophical origins in [1,2], where it was argued that:

1. to have a meaning is to have one from among a set of possible meanings;
2. giving meaning is the process of choosing or *preferring* among those;
3. meanings are, if anything, only other symbols, usually the words of the *explanations we give to convey* meanings to others; and
4. referential accounts of meaning receive too much attention, since we hardly ever point at anything in order to give, or clarify, meaning in adult communication.

The first and second assumptions have a procedural flavor and only the third a representational one: PS is primarily a procedural theory but it has nonetheless been strongly associated with a particular kind of network representation.

PS was developed as a computational semantic theory of natural language. The main topic addressed has always been semantic ambiguity, lexical ambiguity, case ambiguity, and anaphoric ambiguity in line with the first and second assumptions above. Special attention has also been paid to interpreting metaphor, learning new word senses (sense extension) and producing a robust theory that can always produce some interpretation of a sentence. Section 2 describes PS in more detail.

2. PREFERENCE SEMANTICS

PS is a theory of language in which the meaning of a text is represented by a complex semantic structure that is built up out of components; this compositionality is a typical feature of semantic theories. The principal difference between PS and other semantic theories is in the explicit and computational treatment of ambiguous, metaphorical, and non-standard language use. It is an assumption of PS that these features of natural language use are standard and endemic and no theory of language can ignore them.

The original application of PS in the mid 1960s was to the semantic analysis of philosophical texts [3,4]. In the early 1970s, Wilks developed an English-French machine translation system [5]. Since the theory is procedural, rather than representational, it has been possible to provide some empirical tests of its claims and the results have been encouraging: see [6–8], discussed later.

Typeset by $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX

2.1. Main Principles

The main principles of PS, which underlie the sequence of papers [3,5,9,10], are as follows:

(a) *Computational Semantics*

Some of the following principles are part of a general trend within artificial intelligence and computational linguistics that can be called computational semantics [11]—a trend opposed not only to independent syntactic processing and to model-theoretic semantics, but also opposed to a central theme in artificial intelligence that could be styled "naive knowledge-based processing" or, in its extreme form, expert systems.

That extreme form could be identified with the claim that, if adequate knowledge structures are available to a parser, then practical problems of language semantics (lexical ambiguity, etc.) do not arise—the answer to which is that the claim is false, unless the domain chosen is trivial. Computational semantics, however, is not marked off from knowledge realms and their formal expression: on the contrary, knowledge of language and "the world" are not separable, just as they are not separable into data bases called, respectively, dictionaries and encyclopedias.

(b) *Procedural Semantics*

Procedures can give the meaning of quantifiers and other symbols (see [12]). In PS, the description of the representation and the mechanisms that generate it should, ideally, all be procedural and the representation itself should be the product of a few, general and autonomous (not content-dependent) procedures. The procedural semantics idea goes back to the "meaning as use" slogan of Wittgenstein [13] and the operationalist explanations of symbols of Bridgman [14], and is central to the work of, e.g., Winograd [15] and Woods [16].

(c) *Least Effort Principle*

Procedures should be consistent with a Least Effort principle of language understanding [9,17] of doing the minimum amount of analysis necessary, a consequence of which is that linguistic forms, however well-formed on other grounds, which require excessive processing cannot, and should not, be understood.

(d) *Preference*

The appropriate, which is to say the most internally coherent, representation should be chosen for a text from among competing representations. Certain representational structures can be seen as "preferring" other associated representations, and an overall representation for a text is produced by allowing maximal satisfaction or "best fit" of all 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. Consider:

"My car drinks gasoline." (1)

"John ran a mile." (2)

On the PS view, the violation of preferences, such as the preference of 'drinks' for an animate agent in (1) above, and the preference of 'ran' for zero object in (2), is normal and must not be treated as an exceptional matter outside the core of English. Such locutions are statistically so normal, and so easily understood even when wholly novel, that their representation and processing must be part of the central processes of a language understander.

(e) *Semantic Parsing*

This principle, also held by, among others, the Conceptual Dependency (hereafter CD) group at Yale [18–21], is that English can be parsed to a semantic representation without a module devoted explicitly to syntactic analysis, and without traditional syntactic classification of words or sentence components (e.g., N, NP, VP). The necessary generalizations 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.

(f) *Beliefs*

Analysis of the relationships found in text and the representation of that text is a function of the beliefs of the analysis system [22]. Thus, the analysis of (1) depends on what the system believes about drinking and about cars (thus crossing what would be, for many, a semantic-pragmatic boundary) and, likewise, the analysis of (2) depends on beliefs about running and distance (and so similarly for the so-called syntax-semantics distinction and the class of "intransitive verbs").

(g) *Level of Representation*

The level of representation required for natural language understanding is not fixed, but can be at any appropriate level that the representational scheme allows. However, there need be no assumption that all knowledge of the world must be in such a representation: much shallower levels will serve for many purposes.

(h) *Semantic Primitives*

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, even though these primitives may in the end be no different in type from (non-privileged) words of the language (see [23]).

(i) *Linear Boundaries of Language*

The representation emphasizes the linear, rather than the recursive properties of language: its structure therefore emphasizes the linear boundaries of clauses and phrases (with no special role for sentences) as a basis for a surface representation from which progressively deeper representations can be obtained by inference. The representational 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.

(j) *Not Model Theoretic*

The representation need not be of the model theoretic type, and the classic problems of quantification, etc., can be dealt with by special procedures. From the PS point of view, the errors of model theoretic semantics were that it ignored the fact that language always reached other symbols as referents, not unambiguous objects (the latter criticism was made most forcefully by Quine [24]), and that it ignored an essential feature of natural languages: their constant sense extensibility (see [25]).

2.2. Bases of the Theory

PS contains a number of kinds of knowledge representation and process. The following types of knowledge representation are used for language analysis in the theory:

(1) 80–100 semantic "elements" or primitives;
(2) semantic formulas, which represent word senses;
(3) bare templates, which represent basic "messages";
(4) templates, which represent clauses;
(5) paraplates, which represent the senses of prepositions;
(6) common-sense inference rules, which are used for resolving anaphora;
(7) semantic blocks, which represent whole texts;

and, post-1975,

(8) thesaurus structure, which provides hierarchical structure; and
(9) pseudo-texts, which represent both text structure and the contingent, factual, aspect of "frame structures."

Five different kinds of processes were used for language analysis:

(1) template expansion, which evaluates selection restrictions;
(2) preference, which chooses the semantically most dense semantic reading;
(3) TIE routines (see below), which apply paraplates to correct structures corresponding to prepositional phrases;
(4) extractions, which deepen templates by expanding their case subparts and applying common sense inference rules;

and, post-1975,

(5) projection rules, which attempt to interpret metaphor.

The *semantic primitives* (knowledge representation #1) used in the Preference Semantics System were originally derived from a system of biological classification devised by Richens [26].

An appendix to [23] gives the most complete description of the primitives used in PS. One hundred and three primitives are listed there. Of the 103, 15 are "class" primitives which are prefixed by asterisks to distinguish them from the remaining 88, e.g., *ANI (the class of animate entities) includes MAN (the class of human beings).

The primitives divide into four main groups—cases, actions, substantives, and qualifiers plus two "miscellaneous elements." There are 21 case primitives ([23] states that there are 19, but 21 are listed) including SUBJ (the agent case) and OBJE (the object case) and *DIRE (the general direction case). There are 34 action primitives, including GET (obtaining some thing or substance) and GIVE (yielding up some thing or substance). There are 30 substantive primitives consisting of 11 class elements ([23] states "12" but only lists 11) and 19 non-class elements including SPREAD (an extension) and GRAIN (any kind of structure). There are 16 qualifier elements, including GOOD (morally correct or approved), MAL (indicates the sex of the entity qualified is male) and FEM (indicates the sex of the entity qualified is female).

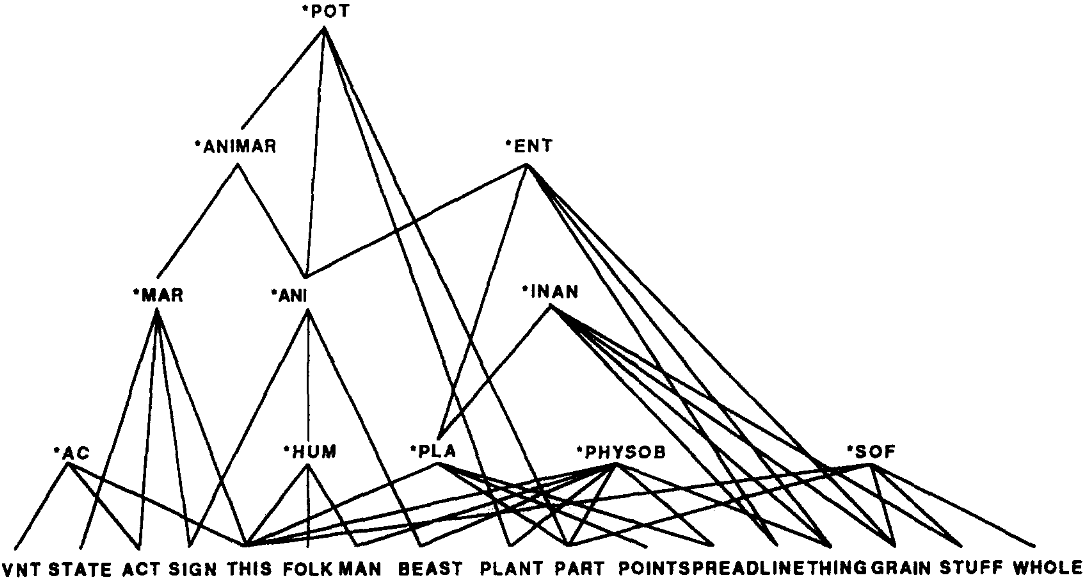

Figure 1. Semantic network of substantive primitives in Preference Semantics (from [27, p. 46]).

The case, substantive, and action primitives are viewable as small, hierarchically organized semantic networks in which the primitives appear as nodes. The network for substantive primitives, which is much more complex than those for the others, is given in Figure 1. The arcs of the network, which are unlabelled, denote class inclusion. Below are definitions of the 11 substantive class primitives from [23].

*POT	any "potent" entity (i.e., *ANI, STATE, PART, ACT, PLANT)
*ANIMAR	*ANI*MAR
*ENT	any entity (i.e., covered by THIS, POINT, FOLK, MAN, GRAIN, PART, THING, BEAST, SIGN, SPREAD, or LINE)
*MAR	any "mark-like" entity (i.e., THIS, SIGN, ACT, STATE)
*ANI	any animate entity (i.e., THIS, MAN, FOLK, BEAST or SIGN)
*INAN	any inanimate entity (i.e., THIS, PART, GRAIN, STUFF, THING, LINE, SPREAD)
*AC	any "act definer" (i.e., THIS, ACT, EVNT)
*HUM	any human entity (i.e., THIS, MAN, FOLK)
*PLA	any "place definer" (i.e., THIS, POINT, SPREAD, PART)
*PHYSOB	any physical object (i.e., THIS, THING, MAN, BEAST, SPREAD, PLANT, PART)
*SOF	any "malleable" entity (i.e., THIS, PART, WHOLE, GRAIN, STUFF)

Semantic formulas (representation #2) are constructed out of semantic primitives and represent individual senses of words. Formulas are constructed from subformulas using a form of dependency syntax. A subformula is any left-right pair whose two members are either a primitive or another subformula. The right-hand element is the governor and the left-hand element the dependent. The very rightmost element of a semantic formula is its ***head***, which represents the "main element" in the sense, e.g., whether a noun refers to a MAN or a THING, or whether a verb denotes an act of THINKing or of DOing.

In a semantic formula the top level subformulas are usually case subformulas which have as their principal (rightmost) elements the case primitives SUBJ, OBJE, GOAL (Purpose) and INST (Instrument).

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

((*ANI SUBJ) (((FLOW STUFF) OBJE) (MOVE CAUSE)))))

One can tell that this formula is for an action because its head, CAUSE, is an action element. Reading the formula: drinking is a causal action, preferably done by animate things (*ANI SUBJ) to liquids ((FLOW STUFF) OBJE). The SUBJ case displays the preferred agent of an action, and the OBJE case the preferred object or patient. Below is the semantic formula for the action 'fire at':

((*HUM SUBJ) ((*ANI OBJE) ((STRIK GOAL) ((THING INST)
((THING MOVE)
CAUSE)))))

It is an action, preferably done by human things, which uses an instrument (a gun) to cause a thing (a bullet) to move, with the goal of striking an animate thing.

Later, Wilks [28] argued that, since no real distinction existed between primitives and words (the former being just a privileged subset of the latter) there could be "mixed-type" representations in which English words are inserted into semantic formulas. For example, the words 'gun' and 'bullet' are inserted into the semantic formula for 'fire at' to give:

((*HUM SUBJ) ((*ANI OBJE) ((STRIK GOAL) ((gun INST) ((bullet MOVE) CAUSE)))))

These formulas were described as "re-entrant" because formulas are mentioned inside each other, i.e., 'gun' and 'bullet' have their own semantic formulas elsewhere in the system. Wilks saw these "mixed-type" formulas as a notational convenience:

> " ... in that 'gun' in a formula for 'shoot' is now just a shorthand form for the formula for 'gun' existing elsewhere in the system. This makes the formulas easier to read for a human user, by avoiding the insertion of too much material in terms of primitives into the body of the formula itself" [29].

A *bare template* (representation #3) is a structure containing the "gist" of a message. Each bare template contains three primitives, an agent-action-object triple (once again, a case-like structure). There is a list of these bare templates, corresponding to numerous basic message forms.

"The crook drank some beer." (3)

In analyzing (3), the heads are obtained for the semantic formulas of 'crook,' 'drink' and 'beer.' Suppose 'crook' has two senses, the less probable one being a shepherd's crook or staff, with the heads [crook (MAN)] and [crook (THING)]. 'Drink' has the head CAUSE and 'beer' has the head (FLOW STUFF). Two candidate interpretations are produced:

[MAN CAUSE (FLOW STUFF)]
[THING CAUSE (FLOW STUFF)]

These are compared against the list of bare templates and, should one not match, it is rejected. Those structures that remain are expanded into templates. The origin of such "gist" structures is to be found in Masterman's work [30].

A *template* (representation #4) is a structure, based on agent, action, and object slots for three semantic formulas that can themselves have dependent formulas, such that the whole structure represents a possible "message," and can contain any number of formulas. Each fragment of a sentence (clause or phrase) has a corresponding template; the existence of more than one template per fragment is representational ambiguity. That ambiguity is reduced by examining 1) the internal "fit" of templates, and 2) the ties between templates for successive fragments of text.

The *template expansion algorithm* (process #1) seeks to resolve this. It consults the formulas in each template and looks at their case subparts to see if their preferences are satisfied. (In what follows, [square brackets] denote the semantic formula for a word-sense. So, for example, [crook(MAN)] denotes the formula for the human sense of the word 'crook' where MAN is the head primitive.) [drink] prefers an animate thing as agent. [crook (MAN)] can satisfy this preference but [crook (THING)] cannot. [drink] prefers a liquid as object; this is available in both representations.

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

[crook (MAN)] → [drink] ← [beer (FLOW STUFF)]
[crook (THING)] → [drink] ← [beer (FLOW STUFF)]

The first of these has the largest number of satisfied preferences, or greater "semantic density," so it is chosen because the process of *preference* (process #2) chooses the template with the most satisfied preferences, i.e., the semantically most dense semantic reading available.

So, a template is a simple graph of three nodes, where any number of other nodes can depend on each of the three. Each node (original or dependent) can be expanded as a formula tree. A certain amount of ingenuity was required to force clauses and phrases of English, say, into this format: prepositional phrases by convention had the sense of the preposition functioning at the central, action, node accompanied by a dummy agent.

```
Triples Notation               Predicate Notation
(CLYDE, ISA, ELEPHANT)         isa (clyde, elephant)
(CLYDE, COLOR, GRAY)           color(clyde, gray)
(CLYDE, LEGS, 4)               legs(clyde, 4)


Network Notation               Frame notation
                               [clyde,
clyde ---isa--- elephant        [[isa, elephant],
clyde ---color--- gray           [color, gray],
clyde ---legs--- 4               [legs, 4]]]
```

Figure 2. Identical information represented in four notations.

In a sense, the underlying triple structure of the template is just another notational variant on predicates, semantic networks, frames, etc., as shown in Figure 2. In the notations, the "relational" primitive (ISA, COLOR, or LEGS) occupies, respectively, the middle element of the triple, the function name of the predicate, the arc label of the network, and the slot-name of the frame.

Paraplates (representation #5) are used to represent the case ambiguity of prepositions. Paraplates are essentially patterns that span two templates, ordered from specific to general.

"He left Comano by the autostrada." (4)
"He left Comano by car." (5)
"He left Comano by following the arrows." (6)

For example, the preposition 'by' expresses various case relations in the examples above. In (4), 'by' is DIRECTION; in (5), the INSTRUMENT case; and in (6), it is the DIRECTION case again.

(*ANI) (MOVE) (WHERE POINT) → [ ] [ ] (WHERE LINE)
(*ANI) (MOVE) (WHERE POINT) → [ ] [ ] (*REAL)
(*ANI) (MOVE) (WHERE POINT) → [ ] (*DO) (WHERE SIGN)

Sentences (4) to (6) have the above paraplates mapped to them, where [ ] represents a dummy agent. Each paraplate can be thought of as expressing a case (those given above). For each paraplate, the three elements before the arrow are mapped onto the template representing a clause preceding the preposition, and all these paraplates match the first clause of each of (4), (5) and (6). The three elements following the arrow are mapped onto the template representing the clause following the preposition.

TIE routines (process #3) are used to apply the paraplates. The routines attempt to "tie" the templates, for clauses surrounding a preposition, to the ordered stack of paraplates for that preposition, until a paraplate is found that is general enough to match. The paraplates are applied as a stack ordered, as mentioned previously, from specific at the top to general at the bottom. The top one is always applied first. The paraplate found determines the case between the main clause and prepositional phrase.

"The soldiers fired at the women and I saw several of them fall." (7)

The problem in (7) is to resolve the ambiguity of 'them' as women rather than soldiers. In such cases, where the tie routines will not resolve the anaphora, *common sense inference rules* (representation #6) are used. The one needed here is:

[1 strikes animate2] → [animate2 falls].

The process of *extraction* (process #4) is used here. It operates on template representations and infers new template forms from the case subparts of formulas. For (7), a template form [soldiers strike women] is produced by deepening the formula for 'fire at': women being the OBJEct of the firing action. [soldiers strike women] matches the left hand side of the common sense inference rule above, hence, the inference that the women fell.

A *semantic block* (representation #7) represents a paragraph (or even a whole text). It consists of templates bound together by paraplates and common sense inferences.

After 1975, a *thesaurus* (representation #8) structure was added because it "provides the lexical specificity that primitive representations lack" ([29, p. 107]). Wilks [10] describes his idea of a thesaurus in some detail. A thesaurus, like Roget's, is a grouping of English words into semi-synonymous rows, usually having the same part of speech type. These rows are grouped under one of about a thousand heads, which in turn are grouped under about ten very general sections. These general Roget heads could be identified with such primitives as MAN, THING, and perhaps other primitives corresponding to the section names from Roget's Thesaurus, e.g., #Abstract Relations (GRAIN), #Space (WHERE), #Matter (STUFF), #Intellect (THINK), #Volition (GOAL), etc. [10, p. 215].

Under the very general section #Volition we might find the head, say #22 propulsion, and under that we would find a subhead #221 firer, attached to some row of "firer" words:

#221 firer: gun, bow, rifle, howitzer ...

Similarly, row #222 projectile might name projectiles, e.g., bullet, arrow.

Heads, subheads and row members all have associated semantic formulas. Row-co-members, e.g., 'gun' and 'bow' should have common parts to their formulas, e.g., all are THINGS, all have a GOAL of hitting something. This common part should be the simpler, more general formula for that row's subhead, #firer. This progressive generalization should extend right up the thesaurus to the general section names [10, p. 205]. The thesaurus imposes a hierarchy of formulas and pseudo-texts, where row members will have structurally similar pseudo-texts.

The *pseudo-text* (representation #9) is Wilks' conception of a frame. "A pseudo-text is a structure of factual and functional information about a concept or item" [10, p. 208]. A pseudo-text is composed of templates linked by case ties, exactly the case ties that the TIE routines above would impose. Hence, a pseudo text is not only a frame-like knowledge structure, but is also identical to the representation that a PS parser would output. On this view, as noted at the beginning, precompiled knowledge structures and those derived from texts are of the same type.

A pseudo-text for 'car' is shown in [10, p. 208]. Its first two lines refer to the insertion of fuel; the next three to the fact that the gasoline-using engine moves the car; and the last four to the way the driver turning the wheel changes the direction of the car. This entity is pointed to by the token "car" which also points to the semantic formula [car].

The pseudo-texts for general primitives like MAN (which could then be equated to very general heads at the top of the thesaurus, in this case "Humanity") would consist of very general assertional forms like:

MAN HAVE THING
MAN THINK SIGN
MAN WANT THING

which are also exactly the forms of bare templates (see above).

> "Readers will have remarked that the whole formula/pseudo-text distinction rests on some intuitive meaning/factual distinction that cannot be formally justified ... I think one can only say that the meaning/factual distinction, even if not philosophically sound, does have some role in our understanding" [10, p. 221].

By contrast, Charniak's frames [31] put both formula-like and fact-like information in the same formalism, although a reader can easily detect which is which. The notion of connecting the generality expressed by the primitives with the generality of taxonomy to be found in the "upper" heads of a thesaurus is also due to Masterman [32].

The process of *projection* (process #5) operates on pseudo-texts. A brief example of projection is given for sentence (1). Projection operates only on preference violations. The best representation of (1) contains a preference violation ("car drinks"), so projection is used. The algorithm compares the template representation

[my+car drink gasoline]

for (1) against templates from the pseudo-text of 'car' seeking the closest match, and selects [ICengine (USE) #liquid] because a car has an IC (internal combustion) engine and gasoline is a liquid. (USE) is then projected onto "drink" in the sentence representation, which becomes

[my+car use gasoline].

In a later development of PS, pseudo-texts were used as the representational basis for a theory of beliefs [22,33,34]: they became structures (renamed environments) that were to be considered as the set of beliefs of an individual about a topic, which might in turn be another believer. An algorithm was given for nesting such belief environments so as to produce default structures for the beliefs of believers by inference, i.e., not known directly. The only real connection to PS was an argument that, since such nested belief structures were required for understanding the utterances and texts of others, only some such default projection would be plausible as a least-effort realization of what would otherwise be an over-massive computational enterprise. It is argued strongly [35] at this stage that belief is essentially involved in natural language understanding, and that many of the phenomena discussed above (e.g., metaphoricity, prepositional phrase attachment, differences in meanings and anaphoric reference across individuals) should be seen as differences of belief.

In another later development [36], an algorithm was given that established the ties binding templates for prepositional phrases into the semantic block, based not on paraplates but on the preferences of the principal words of the sentence: nouns, verbs and prepositions. PS was never properly extended to handle phrase attachment until this 1985 "walk through" algorithm. The algorithm located the best structure by a two-pass processing of a sentence, once leftwards and once rightwards. That alone showed that it was not a plausible psychological process, but that has never been a principal aim of PS. In a test against other methods of attachment [6] this algorithm performed the best.

Readers may have noticed that in the preceding description of PS, the term "preference" had several different but related meanings. Two different meanings of "preference" were noted in [37]. The first meaning of "preference" is as a *selection restriction* in the manner of [38]. These preferences are found in semantic formulas (representation #2) and paraplates (representation #5) and, in a weak form, in bare templates (representation #4). Hence, in the semantic formula for 'drink' shown earlier, its agent preference is the primitive *ANI and the object preference is (FLOW STUFF).

The second meaning of "preference" is as a *boolean value*, either satisfied or violated. In the Preference Semantics System preferences are evaluated by the template expansion algorithm (process #1) operating on the hierarchies of substantive, action, and case primitives (representation #1). If those hierarchies are viewed as semantic networks with arcs denoting class inclusion, the algorithm checks to see if preferences are satisfied by constructing network paths: satisfied preferences are those paths through the semantic networks of primitives that denote instances of class inclusion. Correspondingly, violated preferences are network paths that denote class exclusion.

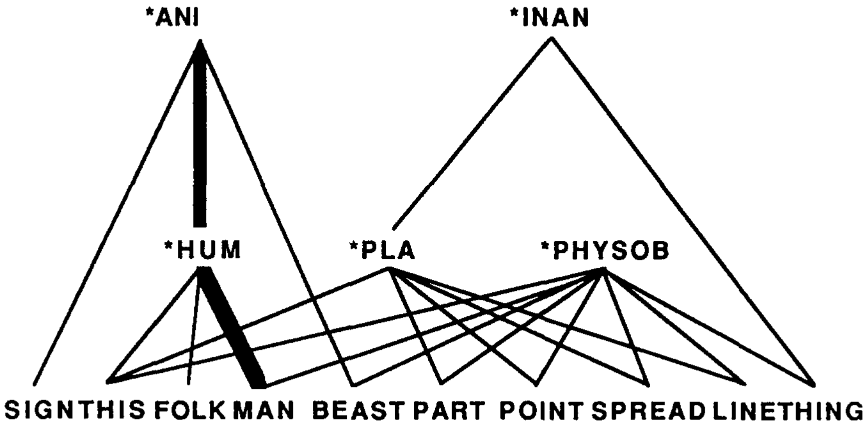

Figure 3. A satisfied preference (i.e., a semantic network path denoting inclusion).

To illustrate this, consider the preceding analysis of (3) and the satisfied preference found between the agent preference of 'drink' and the human sense of 'crook.' The agent preference for [drink] is for *ANI, the class of animate things, and the head primitive of the thief sense of 'crook' is MAN. These two primitives, *ANI and MAN, are shown in Figure 3, which is part of the semantic network of substantive primitives shown in Figure 1. The network path between *ANI and MAN is highlighted in bold. The path denotes inclusion, i.e., men are animate things, hence the preference *ANI is satisfied.

A third meaning of "preference" is *selecting the representation with the greatest semantic density*, which in PS is the representation with the greatest sum of satisfied preferences. The process of preference (process #2) embodies this notion of preference, which is the one meant by the term "Preference Semantics" and the one mentioned as principle (d) underlying PS in Section 2.1.

A fourth meaning of "preference" is *selecting the most specific representation*: alternatives are ranked from specific to general, and each one is tried in turn until a match is found. In PS, paraplates (representation #5) are ordered and tried in this way by the TIE routines (process #3). To quote Wilks [39, p. 165], "paraplates, for a given preposition, are ordered and 'more preferred template' simply means 'the paraplate applied earlier'."

3. EXTENSIONS OF PREFERENCE SEMANTICS

Students of Wilks and other researchers have developed the theory in a number of directions. Boguraev's work [40] and his analyzer has had a strong influence on the research of Alshawi [41], Carter [42,43], Cater [44–46], Fass [27], Huang [47,48] and Tait [49–51] though, by adding an explicit syntactic analysis component and depth-first syntax-driven semantic interpretation, all of this work departs from Wilks' views on semantic parsing. Slator's [52,53] PREMO analysis system, which was less influenced by Boguraev, is most in the spirit of the original PS program. Grishman and Sterling's [8] implementation of PS is also described.

Boguraev [40] focussed on paraphrase, lexical case, and structural ambiguity [54]. He simplified some of the representations and processes: introduced obligatory cases into semantic formulas, removed bare templates, and simplified paraplates into preplates. Moreover, he put in a depth-first syntax-driven (ATN) semantic analyzer in which PS was used to check the internal semantic consistency within syntactic constituents.

Boguraev's semantic analyzer was integrated into a natural language interface to a database [55–59]. The same analyzer was also the basis of a series of parsers that used modified versions of PS (Alshawi, Carter, Cater, Fass, Huang, Tait) and that addressed a subset of the semantic problems tackled by PS. All of these parsers followed Boguraev's lead in being syntax-driven. Alshawi used Boguraev's parser and focussed on lexical ambiguity resolution and inference. Carter's contributions were in the areas of improved anaphor resolution and more systematic semantic formulas, including a better form of self-reference than Wilks' use of the primitive SELF. Cater developed an ATN-based semantic analyzer that used a CD-style of semantic representation together with semantic interpretation based on the use of preferences. Huang developed XTRA, a large English-to-Chinese machine translation system. Semantic interpretation in XTRA was based on simple type checking. Metaphor handling was done using relaxation techniques. Special attention was paid to the resolution of structural ambiguity [60] and language generation [61]. Huang and Guthrie [62] explored implementing a parallel processing version of XTRA.

Fass's Collative Semantics (hereafter CS) is modeled on PS and aims to provide a deeper semantic analysis of metaphor, metonymy, and other related semantic problems. The focus of the theory is on lexical ambiguity and seven kinds of *semantic relations*: literal, metonymic, metaphorical, anomalous, redundant, inconsistent, and novel relations.

"The *kettle* is boiling" (= the liquid in the kettle) (8)

"Denise drank the *bottle*" (= the liquid in the bottle). (9)

In a metonymy, the name of one thing is substituted for that of another related to it. Sentences (8) and (9) contain examples of metonymy in which the name of a container (kettle, bottle) is substituted for its contents. Both of these are CONTAINER FOR CONTENTS metonymies. These examples can be contrasted with (1) which contains a metaphor. The difference between metaphor and metonymy, according to CS, is that the core of a metaphor is a relevant analogy whereas the core of a metonymy is a chain of one or more "metonymic concepts" [63], like CONTAINER FOR CONTENTS or PART FOR WHOLE. For more on this, see [64,65].

CS has four components, consisting of two representations, *sense-frames* and *semantic vectors*, and two processes, *collation* and *screening*. Sense-frames are the knowledge representation scheme and represent individual word senses. Collation matches the sense-frames of two word senses and discriminates the semantic relations between the word senses, as a complex system of mappings between their sense-frames. Semantic vectors represent the systems of mappings produced by collation and hence, the semantic relations encoded in those mappings (except for metonymic relations). Screening chooses between two semantic vectors by applying rank orderings among semantic relations and a measure of conceptual similarity, thereby resolving lexical ambiguity.

CS has been implemented in a computer program called meta5, which analyzes sentences, discriminates the seven kinds of semantic relation between pairs of word senses in those sentences, and resolves any lexical ambiguity. The meta5 program consists of a lexicon containing the sense-frames of about 500 word senses, a small grammar, and semantic routines that contain the collation and screening processes.

Sense-frames are an extension of Wilks' "mixed-type," "re-entrant" semantic formulas but, instead of containing a mixture of primitives and words, sense-frames are composed of word senses which perform the function of the semantic primitives in PS. These word senses are "re-entrant" because they have their own sense-frames, much like Quillian's [66] planes, and much like the circular organization of a real dictionary.

Sense-frames contain *preferences* and *assertions*, which distinguish two uses of semantic information. A preference contains semantic information expressing a restriction on the local context [9,37]; an assertion contains semantic information to be imposed onto the local context. For example, the adjective 'female' has a preference for an animate thing (e.g., an animal or plant), but its assertion is that the same animate thing is female, hence a "female person" is a person (an animate thing) that is female.

Slator's [52,53] PREMO, short for PREference Machine Organization, is a robust knowledge-based parser for natural language. Parsing is robust in that some structure is returned for every input, no matter how ill-formed or "garden-pathological" it is. PREMO contains a *phrase grammar* and a *machine-readable lexicon* derived from the Longman Dictionary of Contemporary English [67], hereafter LDOCE. The derivation of the lexicon is described elsewhere [11,53,68–72]. PREMO uses not only the dictionary entries from LDOCE, but also the grammatical predictions that are supplied, the type hierarchy (as given), and the pragmatic hierarchy (somewhat restructured). This contrasts with the original PS system which used a hand-coded lexicon and which had very little other knowledge of either grammar or pragmatics.

PREMO is organized like the stack model of a standard operating system. Partial parses are represented in "process control block" structures called *language objects*. The control structure is a priority queue of these language objects. The language object at the front of the queue has the highest score, as computed by a preference metric that weighs grammatical predictions, semantic type matching and pragmatic coherence. Each parse is a process operating on a private copy of the current sentence. The first process to emerge from the queue with its sentence buffer empty is declared the winner and is saved.

Every language object receives a preliminary preference score when first created. The initial score depends on the word's sense number (the lower the word sense number, the higher the score, thereby giving an advantage to the lower numbered and more commonly used word senses). Immediately thereafter, various attributes of the language object are evaluated and the initial score is adjusted. Adjustments to scores are either "minor" (2%), "standard" (10%), or "major" (50%), and can be in either direction. Attributes that cause preliminary scores to be *decreased* include an LDOCE time-and-frequency code of "archaic" or "rare." Preliminary scores are slightly *increased* if, for example, the language object is for a phrasal definition (such as "alternating current"), or if it makes a grammatical prediction, or if it is a word with only a single sense definition. Finally, scores are strongly influenced, in either direction, by the position of the word with respect to a restructured pragmatic hierarchy computed for the text. For example, if the text has a scientific orientation, then the scientific senses of words are given increased scores (such as the scientific senses of "measure"), and the scores of other senses of those words are decreased (such as the political and musical senses, of "measure").

The input to PREMO is an unconstrained text and a collection of lexical semantic objects, one for every sense of every word in the text [71]. Each lexical semantic object contains grammatical and sub-categorization information, often with general (and sometimes specific) grammatical predictions; content word objects also have semantic selection codes; many have contextual (pragmatic) knowledge with a score for comparing word senses for their contextual coherence, and the text of selected dictionary definitions.

PREMO employs a uniform representation at the word, phrase, and sentence levels. Further, at every step in the process there is a dominating language object visible; that is, there is always a "well-formed partial parse" extant. This gives an appealing processing model (of a language understander that stands ready to accept the next word, whatever it may be), and a real-time flavor, where the next word is understood in the context of existing structure. One of the PREMO design principles is "always return something" and that policy is guaranteed by keeping every possibility open, if unexplored (this is the PREMO approximation to back-tracking).

By exploiting the operating system metaphor for control, PREMO inherits some very attractive features. First, PREMO avoids combinatorial explosion by ordering the potential parse paths and only pursuing the one that seems the best. This is antithetical to the operating system principle of "fairness," a point where the metaphor is intentionally abandoned in favor of a scheme that has some faint traces of psychological plausibility. Second, the operating system metaphor is an extendible one; that is, it is possible to conceive of PREMO actually being implemented on a dedicated machine. Further, since the multiplication factor at each cycle through the algorithm is small (in the 40–60 range for the near-worst case of 10–12 word senses, times 4–5 applicable grammar rules), and since each of these pairings is independent, it is easy to imagine PREMO implemented on a shared memory parallel processor (like a Sequent). Each of the pairs would be dispatched to a different processing element.

"Fred licked his rifle all over and found that the stock tasted good." (10)

One can view Slator's work as dealing with a criticism of PS, best put by Phil Hayes [73] who argued that in (10) the *local* preferences of PS would always resolve "stock" to "soup" rather than the (correct) "gun part." Slator's PREMO has shown how to meld local and global preferences so as to tackle precisely such examples.

Grishman and Sterling [8] have recently implemented a version of PS as part of the PROTEUS message understanding system, which processes US Navy OPREP (OPeration REPort) messages concerning sightings and engagements at sea. Using PS, PROTEUS correctly analyzed 74% of a corpus of 105 messages; without PS, only 54% were correctly analyzed.

4. RELATIONSHIP TO OTHER WORK

This section relates PS family work against related work, using some of the principles described in Section 2.1.

(a) *Computational Semantics*

Among the approaches to computational semantics, PS has been most closely associated, and confused, with Conceptual Dependency theory or CD (see [18]). There has certainly been cross influence with CD work. Lehnert [74] added noun representations to CD which had until then been wholly verb-based. The PHRAN parser [75] explicitly added template-like triples of atoms to its parsing recognition system. Carbonell's work on anaphora resolution [76] and robust parsing [77] uses an explicit preference measuring algorithm. In the other direction, the addition of the common-sense inference rules to PS in 1974 was almost certainly under the influence of inference rules in Rieger's [78] work.

Schank and his associates always took a much less relativistic view of primitives, believing that there really was a correct set and that this had a hardware implementation in human brains. On that view, the primitives were not special *words* at all, as on the PS view they plainly were. This was an attempt within CD to achieve successful extra-linguistic reference (even if only to bits of the brain), an idea that makes little sense within PS.

A remaining difference has been the importance of psychological claims in the CD school and their absence in PS, as witness, the two-pass attachment algorithm of [36] which could hardly be psychologically valid, but which is effective, and the claimed symmetry of analysis and generation in PS [79] whereas CD researchers have always strongly emphasized any connection between their representations and psycholinguistic evidence.

The emphasis in the CD group shifted in the 1980s, from describing their work as natural language processing, to describing it as human memory research, but the change was largely cosmetic, since at Yale the approach was still applied to machine translation and data-base front-ends. Lehnert's [80] recent work, although its title suggests a continuing concern with memory, is very much a return to early CD concerns and is certainly computational semantics: in fact, a fusion of PS and CD approaches.

It is also connectionist as well, by which one means the current cluster of artificial intelligence theories around the notion of small computing units, connected in very large numbers, "learning from experience" by means of shifting aggregated weights in a network (see [81,82]). This development shares many of the views of PS: an emphasis on non-syntactic methods, a continuity with other forms of world knowledge, and some difficulty in reconciling its claims with those of logic-based approaches.

(d) *Preference*

Preference-like notions have been stressed in connectionism, such as competition between representational structures, where the stronger, more connected, ones "win out." Connectionism, however, seeks a more general algorithm than the simple counting relativity by which PS establishes the densest network. Sparck Jones (personal communication) has made the point that preference counting was less plausible if it were 7 over 6 (as opposed to 10 over 1). This ties directly to the concerns of connectionists, who are much concerned with generating "a big enough gap" over their representations.

Preference notions can also be seen in the general idea of optimizing numerical values for lexical ambiguity resolution, in the marker passing approaches of Charniak [83] and Hirst [84], as well as connectionist approaches to lexical ambiguity resolution, such as Waltz and Pollack's [82].

Grishman and Sterling's [8] version of PS included a "longest parse" mechanism which was built into their original scoring scheme. Their notion of partial model is a key one here: a formal technique that leads to algorithms of exactly this type (selecting the largest partial model). In the belief area, this is not so far from Gaspar's [85] notion of preferred model of belief (which tends to be the largest set though her method of choosing it is unclear).

Other related notions of preference have appeared over the years in natural language processing, often used for ambiguity resolution, including [86–91]. Preference-like notions also appear in robust parsing techniques, for example, "least-deviant first parsing" [77] resembles the most-preferred first parsing used in PS.

A notion of "syntactic preference" has been used by Pereira [92], Hobbs and Bear [93] and others, from the Frazier and Fodor [94] tradition of Minimal Attachment and Kimball's [95] ideas on Right Association, and this tradition has now adopted the notion of "lexical preference," possibly stemming from PS (see also Schubert's work [96,97]).

Pereira [92] proposes what he claims is a simple, precise, falsifiable, general framework "in which improved versions of Right Association and Minimal Attachment can be formulated." Within the framework, the two principles RA and MA "correspond to two precise rules on how to choose between alternative parsing actions." It is also claimed that principles like RA and MA can be described "starting from very few assumptions about grammar and parsing mechanisms." The parsing mechanism proposed is bottom-up using the shift-reduce method. Hobbs and Bear's [93] approach within their Dialogic system is generalized to "pick the reading with the most restrictive context."

(e) *Semantic Parsing*

There has always been confusion about what was semantic parsing as part of PS. It was certainly never claimed that syntactic information was not used in parsing: it is only necessary to note that a parser of English needs to know that adjectives normally follow determiners, to show that semantic parsing proponents could hardly have denied that. What semantic parsing argued was that all such information could be expressed in semantic terms, whether or not it was more efficient to do so, and so no separate syntactic analysis module was required.

What semantic parsing is certainly inconsistent with is any syntactic parsing theory that claims (e.g., [98]) that the constraints of Chomsky's Universal Grammar can be given procedural form, and that there are important generalizations, having nothing whatever to do with meaning, that any parsing system must take account of, and which cannot be expressed at all in an semantic parsing approach.

"The lecture was believed to have been given yesterday." (11)

The work of Marcus and others has marvellously sharpened what it is semantic parsing has to show to stay in business. Consider the Marcus/trace account of a sentence like (11). Any semantic parsing account can deal with this, given plausible assumptions about the semantic preferences for filling the slots involved, e.g., that "believe" prefers a clause to an entity as object, if it can get one, etc., taken together with general heuristics about what kinds of slot (e.g., subjects) hold or cede their contents during any reorganization of the structure of the sentence.

(h) *Semantic Primitives*

There has been much misunderstanding of the claims about primitives in PS and their relativistic nature (as opposed to the CD claims). Criticisms of the notion of primitives from Hayes [99] and Winograd [100] have missed this point, as did the architects of KRL [101] when attacking PS, an issue that was clarified in [102].

REFERENCES

1. Y.A. Wilks, Decidability and natural language, *Mind* 80 (4), 497–520 (1971).
2. Y.A. Wilks, *Grammar, Meaning and the Machine Analysis of Language*, Routledge and Kegan Paul, London, (1972).
3. Y.A. Wilks, Argument and proof in metaphysics, from an empirical point of view, Ph.D. Thesis, Cambridge, England (1968).
4. Y.A. Wilks, On-line semantic analysis of English texts, *Machine Translation* 11 (3/4), 59–72 (1968).
5. Y.A. Wilks, An artificial intelligence approach to machine translation, In *Computer Models of Thought and Language*, (Edited by R.C. Schank and K.M. Colby), pp. 114–151, W.H. Freeman, San Francisco, CA, (1973).
6. G. Whittemore, K. Ferrara and H. Brunner, An empirical study of the predictive powers of simple attachment schemes for post-modifier prepositional phrases, *Proceedings of the 28th Annual Meeting of the ACL*, Pittsburgh, PA, pp. 23–30, (1990).
7. R. Grishman and J. Sterling, Analyzing telegraphic messages, *Proceedings of the DARPA Speech and Natural Language Workshop*, February 21–23, 1989, pp. 204–208, Morgan Kaufmann, Los Altos, CA, (1989).
8. R. Grishman and J. Sterling, Preference semantics for message understanding, *Proceedings of DARPA Workshop*, Cape Cod, MA (1989).
9. Y.A. Wilks, An intelligent analyzer and understander for English, *Communications of the ACM* 18 (5), 264–274. Reprinted in *Readings in Natural Language Processing*, (Edited by B.J. Grosz, K.S. Jones and B.L. Webber), pp. 193–203, Morgan Kaufmann, Los Altos, CA (1975).
10. Y.A. Wilks, Making preferences more active, *Artificial Intelligence* 11 (2), 197–223 (1978). Reprinted in *Associative Networks: Representation and Use of Knowledge by Computers*, (Edited by N.V. Findler), pp. 239–266, Academic Press, New York, NY, (1979).
11. Y.A. Wilks, D.C. Fass, C-M. Guo, J.E. McDonald, T. Plate and B.M. Slator, A tractable machine dictionary as a resource for computational semantics, In *Computational Lexicography for Natural Language Processing* (Edited by B.K. Boguraev and T. Briscoe), pp. 193–228, Longman, Harlow, England, (1989).
12. Y.A. Wilks, Some thoughts on procedural semantics, In *Strategies for Natural Language Processing*, pp. 495–516, Lawrence Erlbaum Associates, Hillsdale, NJ, (1982).
13. L. Wittgenstein, *Philosophical Investigations*, Blackwell, Oxford, England, (1953).
14. P. Bridgman, *The Nature of Physical Theory*, Princeton University Press, Princeton, NJ, (1936).
15. T. Winograd, *Understanding Natural Language*, Academic Press, New York, NY, (1972).
16. W.A. Woods, An experimental parsing system for transition network grammars, In *Natural Language Processing*, (Edited by R. Rustin), pp. 112–154, Algorithmics Press, New York, NY, (1973).
17. Y.A. Wilks, A preferential pattern-seeking semantics for natural language inference, *Artificial Intelligence* 6 (1), 53–74 (1975).
18. S. Lytinen, Conceptual Dependency and its descendants, (this volume).
19. R.C. Schank, *Conceptual Information Processing*, Amsterdam, North-Holland, (1975).
20. R.C. Schank and R.P. Abelson, *Scripts, Plans, Goals and Understanding*, Lawrence Erlbaum Associates, Hillsdale, NJ, (1977).
21. R.C. Schank and C.K. Riesbeck, *Inside Computer Understanding: Five Programs Plus Miniatures*, Lawrence Erlbaum Associates, Hillsdale, NJ, (1981).
22. Y.A. Wilks and A. Ballim, Multiple agents and the heuristic ascription of belief, *Proceedings of the 10th International Joint Conference on Artificial Intelligence (IJCAI-87)*, Milan, Italy, pp. 118–124 (1987).
23. Y.A. Wilks, Good and bad arguments about semantic primitives, *Communication and Cognition* 10 (2), 181–221 (1977).
24. W.V.O. Quine, *Word and Object*, MIT Press, Cambridge, MA, (1960).
25. Y.A. Wilks, On keeping logic in its place, In *Theoretical Issues in Natural Language Processing*, Lawrence Erlbaum Associates, Hillsdale, NJ, (1989).
26. R.H. Richens, Tigris and Euphrates, *Proceedings of the Symposium on the Mechanisation of Thought Processes*, National Physical Laboratory, Teddington, England, HMSO (1958).
27. D.C. Fass, Collative semantics: A semantics for natural language processing, Ph.D. Thesis, Memorandum in Computer and Cognitive Science, MCCS-88-118, Computing Research Laboratory, New Mexico State University, Las Cruces, NM, (1988).
28. Y.A. Wilks, Primitives and words, *Proceedings of the 1st Workshop on Theoretical Issues in Natural Language Processing (TINLAP-1)*, Cambridge, MA, pp. 38–41 (1975).
29. Y.A. Wilks, Seven theses on artificial intelligence and natural language, Working Paper No. 17, Institut Pour Les Etudes Semantiques et Cognitives, Universite de Genève, Switzerland (1975).

30. M. Masterman, Semantic message detection for machine translation, using an interlingua, *Proceedings of the 1961 International Conference on Machine Translation of Languages and Applied Language Analysis*, National Physical Laboratory, Teddington, Middlesex, England, Her Majesty's Stationery Office, pp. 437–475 (1961).
31. E. Charniak, A framed PAINTING: On the representation of a common sense knowledge fragment, *Cognitive Science* 1 (4), 355–394 (1977).
32. M. Masterman, The thesaurus in syntax and semantics, *Mechanical Translation* 4 (1/2) (1957).
33. Y.A. Wilks and J. Bien, Speech acts and multiple environments, *Proceedings of the 6th International Joint Conference on Artificial Intelligence (IJCAI-79)*, Tokyo, Japan, pp. 968–970 (1979).
34. Y.A. Wilks and J. Bien, Beliefs, points of view, and multiple environments, *Cognitive Science* 7 (2), 95–116 (1983).
35. A. Ballim, Y.A. Wilks and J.A. Barnden, Belief ascription, metaphor, and intensional identification, Accepted for *Cognitive Science.* Also Memoranda in Computer and Cognitive Science MCCS-88-138, Computing Research Laboratory, New Mexico State University, NM (in press).
36. Y.A. Wilks, X. Huang and D.C. Fass, Syntax, preference and right attachment, *Proceedings of the 9th International Joint Conference on Artificial Intelligence (IJCAI-85)*, Los Angeles, CA, pp. 779–784 (1985).
37. D.C. Fass and Y.A. Wilks, Preference semantics, ill-formedness and metaphor, *American Journal of Computational Linguistics,* Special Issue on Ill-Formed Input 9 (3/4), 178–187 (1983).
38. J.J. Katz and P. Postal, *An Integrated Theory of Linguistic Description*, MIT Press, Cambridge, MA, (1964).
39. Y.A. Wilks, Parsing English II, In *Computational Semantics,* (Edited by E. Charniak and Y.A. Wilks), pp. 155–184, North-Holland, Amsterdam, (1976).
40. B.K. Boguraev, Automatic resolution of linguistic ambiguities, Ph.D. Thesis, Technical Report No. 11, University of Cambridge Computer Laboratory, Cambridge, England, (1979).
41. H. Alshawi, *Memory and Context for Language Interpretation*, Cambridge University Press, Cambridge, England, (1987).
42. D.M. Carter, Common-sense inference in a focus-guided anaphora solver, *Journal of Semantics* 4 (4), 237–246 (1986).
43. D.M. Carter, *Interpreting Anaphors in Natural Language Texts*, Ellis Horwood, Chichester, England, (1987).
44. A.W.S. Cater, Analysis and inference for English, Ph.D. Thesis, Technical Report No. 19, University of Cambridge Computer Laboratory, Cambridge, England, (1981).
45. A.W.S. Cater, Request-based parsing with low-level syntactic recognition, In *Automatic Natural Language Parsing,* (Edited by K. Sparck Jones and Y.A. Wilks), pp. 141–147, Ellis Horwood, Chichester, England, (1983).
46. A.W.S. Cater, Preference-directed use of ATN's, *Proceedings of the 7th European Conference on AI (ECAI -86)*, pp. 23–28, Munich, (1986).
47. X. Huang, XTRA: The design and implementation of a fully automatic machine translation system, Ph.D. Thesis, Memoranda in Computer and Cognitive Science MCCS-88-121, Computing Research Laboratory, New Mexico State University, Las Cruces, NM, (1988).
48. X. Huang, Semantic analysis in XTRA, An English-Chinese machine translation system, *Computers and Translation* 3 (2), 101–120 (1988).
49. J.I. Tait, Automatic summarising of English texts, Ph.D. Thesis, Dept. of Computer Science, University of Cambridge, England (1982).
50. J.I. Tait, Semantic parsing and syntactic constraints (Mark IV), In *Automatic Natural Language Parsing*, pp. 169–177, Ellis Horwood, Chichester, England, (1983).
51. J.I. Tait, An English generator for a case-labelled dependency representation, *Proceedings of the 2nd Conference of the European Chapter of the Association for Computational Linguistics*, pp. 194–197, Geneva, Switzerland, (1985).
52. B.M. Slator, Lexical semantics, preference semantics, and the machine analysis of text, Ph.D. Thesis, Memoranda in Computer and Cognitive Science MCCS-88-143, Computing Research Laboratory, New Mexico State University, NM (1988).
53. B.M. Slator, PREMO: The PREference machine organization, *Proceedings of the Third Annual Rocky Mountain Conference on Artificial Intelligence*, Denver, CO, June 13–15, pp. 258–265 (1988).
54. B.K. Boguraev, Recognizing conjunctions with the ATN framework, In *Automatic Natural Language Parsing*, pp. 39–45, Ellis Horwood, Chichester, England, (1983).
55. B.K. Boguraev and K. Sparck Jones, A general semantic analyzer for data base access, *Proceedings of the 7th International Joint Conference on Artificial Intelligence, (IJCAI-81)*, Vancouver, BC, Canada, pp. 443–445 (1981).
56. B.K. Boguraev and K. Sparck Jones, Steps towards natural language to data language translation using general semantic information, *Proceedings of the 5th European Conference on AI (ECAI-82)*, pp. 232–233, Paris, France, (1982).
57. B.K. Boguraev and K. Sparck Jones, How to drive a database front end using general semantic information, *Proceedings of the 1st Conference on Applied Natural Language Processing*, pp. 81–89, Santa Monica, CA, (1983).
58. B.K. Boguraev and K. Sparck Jones, A framework for inference in natural language front ends to databases, Technical Report No. 64, University of Cambridge Computer Laboratory, Cambridge, England (1983).

59. B.K. Boguraev and K. Sparck Jones, A natural language front end to databases with evaluative feedback, In *New Applications of Databases*, (Edited by G. Gardarin and E. Gelenbe), pp. 159–183, Academic Press, New York, NY, (1984).

60. X. Huang, A computational treatment of gapping, right node raising and reduced conjunction, *Proceedings of the 10th International Conference on Computational Linguistics (COLING-84)*, Stanford, CA, pp. 243–246 (1984).

61. X. Huang, The generation of Chinese sentences from the semantic representations of English sentences, *Proceedings of International Conference on Machine Translation*, Cranfield, England (1984).

62. X. Huang and L. Guthrie, Parsing in parallel, *Proceedings of the 11th International Conference on Computational Linguistics (COLING-86)*, Bonn, West Germany, pp. 140–145 (1986).

63. G. Lakoff and M. Johnson, *Metaphors We Live by*, Chicago University Press, London, (1980).

64. D.C. Fass, Metonymy and metaphor: What's the difference?, *Proceedings of the 12th International Conference on Computational Linguistics (COLING-88)*, Budapest, Hungary, pp. 177–181 (1988).

65. D.C. Fass, Met*: A method for discriminating metonymy and metaphor by computer, Accepted for *Computational Linguistics*. Also Technical Report CSS/LCCR TR 89-15, Centre for Systems Science, Simon Fraser University, Burnaby, BC, Canada (to appear).

66. M.R. Quillian, Word concepts: A theory and simulation of some basic semantic capabilities, *Behavioral Science* **12** (3/4), 410–430 (1967). Reprinted in *Readings in Knowledge Representation*, (Edited by R.J. Brachman and H.J. Levesque), pp. 98–118, Morgan Kaufmann, Los Altos, CA, (1985).

67. P. Procter, R.F. Ilson, J. Ayto *et al.*, *Longman Dictionary of Contemporary English*, Longman Group Limited, Harlow, England, (1978).

68. B.M. Slator, Extracting lexical knowledge from dictionary text, *Knowledge Acquisition* **1** (1), 89–112 (1989).

69. B.M. Slator, Extracting lexical knowledge from dictionary text, *SIGART Newsletter* (Special Issue on Knowledge Acquisition) **108**, 173–174 (1989).

70. B.M. Slator, Sense and preference (this volume).

71. B.M. Slator and Y.A. Wilks, Toward semantic structures from dictionary entries, *Proceedings of the Second Annual Rocky Mountain Conference on Artificial Intelligence (RMCAI-87)* Boulder, CO, 85–96. Also Memorandum in Computer and Cognitive Science, MCCS-87-96, Computing Research Laboratory, New Mexico State University, NM (1987).

72. Y.A. Wilks, D.C. Fass, C-M. Guo, J.E. McDonald, T. Plate and B.M. Slator, Machine tractable dictionaries as tools and resources for natural language processing, *Proceedings of the 12th International Conference on Computational Linguistics (COLING-88)*, Budapest, Hungary, pp. 750–755 (1988).

73. Philip J. Hayes, Semantic markers and selectional restrictions, In *Computational Semantics*, (Edited by E. Charniak and Y.A. Wilks), pp. 41–54, North Holland, Amsterdam, Netherlands, (1976).

74. W.G. Lehnert, *The Process of Question Answering*, Lawrence Erlbaum Associates, Hillsdale, NJ, (1978).

75. R. Wilensky and Y. Arens, PHRAN: A knowledge-based natural language understander, *Proceedings of the 18th Annual Meeting of the ACL*, Philadelphia, PA, pp. 117–121 (1980).

76. J.G. Carbonell and R. Brown, Anaphora resolution: A multi-strategy approach, *Proceedings of the 12th International Conference on Computational Linguistics (COLING-88)*, pp. 96–101, Budapest, Hungary, (1988).

77. J.F. Lehman and J.G. Carbonell, Learning the user's language: A step towards automated creation of user models, In *User Models in Dialog Systems*, (Edited by A. Kobsa and W. Wahlster), pp. 163–194, Springer-Verlag, Berlin, West Germany, (1989).

78. C.J. Rieger, Conceptual memory, In *Conceptual Information Processing*, pp. 157–288, North-Holland, Amsterdam, (1975).

79. Y.A. Wilks, On the claimed symmetry of analysis and generation, Memoranda in Computer and Cognitive Science, Computing Research Laboratory, New Mexico State University, Las Cruces, NM (to appear).

80. W.G. Lehnert, Learning to integrate syntax and semantics, *Proceedings of the 4th International Workshop on Machine Learning*, Irvine, CA (1987).

81. P. Smolensky, On the proper treatment of connectionism, *Behavioral and Brain Sciences* **11** (1), 97–129 (1988).

82. D.L. Waltz and J.B. Pollack, Massively parallel parsing: A strongly interactive model of natural language interpretation, *Cognitive Science* **9** (1), 51–74 (1985).

83. E. Charniak, Passing markers: A theory of contextual influence in language comprehension, *Cognitive Science* **7** (2), 171–190 (1983).

84. G.J. Hirst, *Semantic Interpretation and the Resolution of Ambiguity*, Cambridge University Press, Cambridge, (1987).

85. G. Gaspar, Communication and belief changes in a society of agents, *Proceedings of the 5th Rocky Mountain Conference on Artificial Intelligence (RMCAI-90)*, Las Cruces, NM, pp. 319–325 (1990).

86. R.P. Fawcett, A systemic functional approach to selectional restrictions, roles and semantic preferences, Accepted for *Journal of Machine Translation* (to appear).

87. H. Inagaki *et al.*, Sentence disambiguation by domain oriented preference sets, *Proceedings of the 13th International Conference on Computational Linguistics (COLING-90)*, Helsinki, Finland (1990).

88. G. Malnati and P. Paggio, A preference mechanism based on multiple criteria resolution, *Proceedings of the 13th International Conference on Computational Linguistics (COLING-90)*, Helsinki, Finland (1990).

89. D. Petitpierre, S. Krauwer, D. Arnold and G.B. Varile, A model for preference, *Proceedings of the 3rd Conference of the European Chapter of the Association for Computational Linguistics*, Copenhagen, Denmark (1987).
90. J. Tsujii, Y. Muto, Y. Ikeda and M. Nagao, How to get preferred readings in natural language analysis, *Proceedings of the 12th International Conference on Computational Linguistics (COLING-88)*, Budapest, Hungary, pp. 683–687 (1988).
91. R.S. Virkar and J.W. Roach, Pattern-based parsing for word-sense disambiguation, *Proceedings of the 10th Annual Cognitive Science Society Conference*, Montreal, Quebec, Canada, pp. 688–694 (1988).
92. F.C.N. Pereira, A new characterization of attachment preferences, In *Natural Language Parsing*, (Edited by D.R. Dowty, L. Karttunen and A.M. Zwicky), pp. 307–319, Cambridge University Press, Cambridge, England, (1985).
93. J. Hobbs and J. Bear, Two principles of parse preference, *DARPA Workshop*, Fall '89, Los Altos, CA, Morgan Kaufmann, San Mateo, CA (1989).
94. L. Frazier and J.D. Fodor, The sausage machine: A new two-stage parsing model, *Cognition* 6 (4), 291–325 (1978).
95. J. Kimball, Seven principles of surface structure parsing in natural language, *Cognition* 2 (1), 15–47 (1973).
96. L.K. Schubert, On parsing preferences, *Proceedings of the 10th International Conference on Computational Linguistics (COLING-84)*, Stanford, CA, pp. 247–250 (1984).
97. L.K. Schubert, Are there preference trade-offs in attachment decisions?, *Proceedings of the 5th National Conference on Artificial Intelligence (AAAI-86)*, Philadelphia, PA, pp. 601–605 (1986).
98. M.P. Marcus, *A Theory of Syntactic Recognition for Natural Language*, MIT Press, Cambridge, MA, (1980).
99. Patrick J. Hayes, Some problems and non-problems in representation theory, *Proceedings of the AISB Conference*, University of Sussex, England, pp. 63–78 (1974).
100. T. Winograd, On primitives, prototypes, and other semantic anomalies, *Proceedings of the 2nd Workshop on Theoretical Issues in Natural Language Processing (TINLAP-2)*, pp. 25–32, University of Illinois at Champaign-Urbana, IL, (1978).
101. D.G. Bobrow and T. Winograd, An overview of KRL, A knowledge representation language, *Cognitive Science* 1 (1), 3–46 (1977).
102. W.G. Lehnert and Y.A. Wilks, A critical perspective on KRL, *Cognitive Science* 3 (1), 1–28 (1979).