PREMO: parsing by conspicuous lexical consumption†

Brian M. Slator and Yorick Wilks*

Computing Research Laboratory
Box 30001
New Mexico State University
Las Cruces, NM 88003-0001

ABSTRACT

PREMO is a knowledge-based Preference Semantics parser with access to a large, lexical semantic knowledge base and organized along the lines of an operating system. The state of every partial parse is captured in a structure called a *language object*, and the control structure of the preference machine 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. The highest priority language object is the intermediate reading that is currently most preferred (the others are still "alive," but not actively pursued); in this way the preference machine avoids combinatorial explosion by following a "best-first" strategy for parsing. The system has clear extensions into parallel processing.

1. Introduction

PREMO: The **PRE**ference Machine Organization, is an architecture, modelled as an operating system, for parsing natural language. Each "ready" process in the system captures the state of a partial parse in a "process control block" structure called a *language object*. The control structure is a priority queue of competing parses, with priority given to each parse "process" on the basis of a preference semantics evaluation. The "time-slice" for each process is whatever is needed to move forward one word in a local process sentence buffer (where each process operates on a private copy of the current sentence). After every time slice, the preference/priority for the currently "running" parse is re-computed and the language object for that process is returned to the priority queue. The first process to emerge from the queue with its sentence buffer empty is declared the winner and saved. This strategy is both a run-time optimization and an application of the "Least Effort Principle" of intuitively plausible language processing. The parsing is robust in that some structure is returned for every input, no matter how ill-formed or "garden-pathological" it is.

1.1. Language Object Structures

The basic data structure manipulated by the preference machine is called a *language object* (Fig. 1). Each language object is a complex structure containing at least the following attributes:

Name and Type: every object *name* is a word from the text. Object *type* defaults to *word* until a word is found to be part of phrase, then that object is changed to type *phrase*. The *head* of a phrase is the language object with its co-members subordinated to it.

† This research was supported by the New Mexico State University Computing Research Laboratory — grateful acknowledgement is accorded to the members of the CRL Natural Language Group for their continuing interest and support.

* Present address: Department of Computer Science, North Dakota State University, Fargo, ND 58105.

Score: see section 3.1, *Computing Preferences*, below.

Lexical Frame: see section 2.1, *Text Lexicon*, below.

Predictions and Requirements: grammatical predictions and constraints, both as found in the language object itself and as synthesized from subordinated language objects (see section 2.1, *Text Lexicon*, below).

Preceding and Remaining Text: the sentential context, or local sentence buffers; (preceding words are needed to check on grammatical predictions and requirements that can be either forward or backward in the sentence). When the *remaining text* is exhausted, the language object is a candidate sentence representation.

Subordinated Language Objects: subordination refers to the way an NP is subordinated to a PP, an AP is subordinated to an NP, and a phrase element is subordinated to the phrase head. A corresponding label list holds labels associated with each stack element.

name:	technician	(string data type)	filler: a word from the text
type:	word	(string data type)	filler: WORD or phrase: NP, AP, PP, VP, or CL (clause)
score: .	1.75	(short-float data type)	filler: preference/priority value
frame:	(POS . noun) (SENSE . 0) (GRAMMAR . noun/count) (SEM-TYPE . human) (PRAG . occupations)	(association list)	filler: a lexical semantic frame instantiated to some particular word sense of the object named
requirement:	nil	(list of atoms)	filler: codes for grammatical requirements from GRAMMAR slots
prediction:	nil	(list of atoms)	filler: codes for grammatical predictions from GRAMMAR slots
previous:	(The)	(list of words)	filler: previous words in the sentence
remainder:	(measures alternating current with an ammeter)	(list of words)	filler: subsequent words in the sentence
cases:	nil	(list of labels)	filler: case and function labels marking constituent relations
subordinate:	nil	(a LIFO stack of language objects)	filler: language objects that form the state of the parse and are linguistically subordinate to the current language object.

Fig. 1: Language object for "technician" (before coalescing).

1.2. Preference Machine Control

PREMO receives two inputs: a text to be parsed, and a lexicon of semantic objects specific to that text. The algorithm of the preference machine, after loading the priority queue with scored language objects for every sense of the first word in the sentence, is as follows (Fig. 2):

1. **Delete-Max** - retrieve the highest priority language object in the queue. (If the sentence buffer for that object is empty then go to 8).
2. **Get-Lexical** - retrieve the list of sense frames associated with the first word in the sentence buffer within the current language object
3. **Make-Language-Object(s)** - instantiate a new language object for every sense of the new word, with an appropriate initial preference score.
4. **Copy-Language-Object** - create a copy of the current high priority language object to pair with each new language object.
5. **Coalesce** - If more than a single grammar rule applies to the pair, copies of the pair are made. Combine the pairs of language objects, subordinating one to the other.

6. **Compute-Preference** - assign a new priority score to each of the language objects resulting from coalescing pairs.

7. **Enqueue** - insert the coalesced language objects onto the priority queue in preference score order. Go to 1.

8. **Inter-sentential Processes** - save the language object at the front of the priority queue and flush the queue. If there are no more sentences, return; otherwise, read the next sentence in the text and load the priority queue with scored language objects for every sense of the first word in the new sentence. Go to 1.

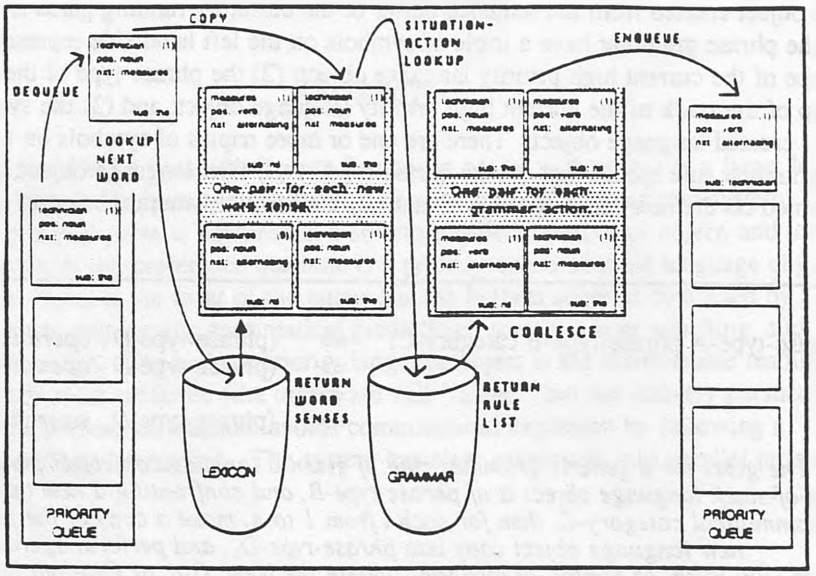
COPY
SITUATION-ACTION LOOKUP
ENQUEUE
DEQUEUE
LOOKUP NEXT WORD
One pair for each new word sense.
One pair for each grammar action.
COALESCE
RETURN WORD SENSES
RETURN RULE LIST
LEXICON
GRAMMAR
PRIORITY QUEUE
PRIORITY QUEUE
technician (1) pos: noun nst: measures sub: the
measures (1) pos: verb nst: alternating sub: nil

Fig. 2: **PREMO:** *the* **PRE***ference Machine Organization.*

2. Global Data

Global system data structures include a *text-specific lexicon*, and a *context structure* derived from Longman's Dictionary of Contemporary English (LDOCE; Procter et al. 1978), and a *phrase grammar*.

2.1. Text Lexicon

LDOCE is a full-sized dictionary in machine-readable form, designed for learners of English as a second language, and containing several non-standard features (grammar, type, and pragmatic codes). A PREMO sub-system produces text-specific lexicons from selected machine-readable dictionary definitions (Wilks, Fass, Guo, McDonald, Plate and Slator, 1987, 1988, 1989). The input to this sub-system is unconstrained text; the output is a collection of lexical semantic objects, one for every *sense* of every word in the text. Each lexical semantic object in this lexicon contains grammatical and sub-categorization information, often with general (and sometimes specific) grammatical predictions; content word objects also have semantic selection codes; and many have contextual (pragmatic) knowledge as well. As a natural side-effect of the lexicon construction, a relative contextual score is computed for each object that bears such a code; these scores provide a simple metric for comparing competing word senses for text-specific contextual coherence, and so directly address the problem of lexical ambiguity. Besides exploiting those special encodings supplied with the dictionary entries, the text of selected dictionary definitions are analyzed, through parsing and pattern matching, to further enrich the resulting representation (Slator, 1988a, 1988b; Slator and Wilks 1987, 1989).

2.2. Syntactic Structures

The PREMO grammatical formalism is non-standard, being not a phrase-structured production system of rewrite rules but rather a phrase-triggered system of situation-action rules. The **Coalesce** procedure lies at the heart of the PREMO algorithm. This routine accepts a pair of language objects:

1. the current high priority language object as retrieved (and perhaps copied) from the priority queue, and
2. a new language object representing a single word sense of the next word in the sentence buffer of the current high priority language object.

Every language object that is retrieved from the priority queue is of type *phrase*, and every new language object created from the sentence buffer of the currently running parse is of type *word*. The rules of the phrase grammar have a triple of symbols on the left hand side representing: (1) the phrase type of the current high priority language object; (2) the phrase type of the language object on the top of the stack of the current high priority language object; and (3) the syntactic category of the newly created language object. There are one or more triples of symbols on the right hand side of each grammar rule specifying: (1) the phrase type of the new language object; (2) the action to be performed on the new language object; and, (3) the location where the action is to take place (Fig. 3).

$$\begin{array}{lll}
\text{(phrase-type-A phrase-type-B category-C)} & \Rightarrow & (\text{phrase-type-}D_1 \; operation_1 \; location_1) \\
 & \Rightarrow & (\text{phrase-type-}D_2 \; operation_2 \; location_2) \\
 & \Rightarrow & \ldots \\
 & \Rightarrow & (\text{phrase-type-}D_n \; operation_n \; location_n)
\end{array}$$

Fig. 3: The gloss for a generic grammar rule: if given a language object of phrase-type-A, whose top-of-stack language object is of phrase-type-B, and confronting a new language object of grammatical category-C, then for each i from 1 to n, make a copy of the pair, change the new language object copy into phrase-type-D_i, and perform operation$_i$ (either Sub, Push, or Subto), at location$_i$ (within the New, Old, or Old-Sub language object).

The set of possible phrase types is limited, at present, to these five: Adjective phrase, Noun phrase, Prepositional phrase, Verb phrase, and Clause (a generic Other phrase type). Although several action triples (all of which get executed), could appear on the right hand side of a rule, in the current implementation no rule exceeds five action triples and most are three or less. Further, the part of speech set in the lexicon derived from LDOCE has 10 members: adjective, adverb, conjunction, determiner, interjection, noun, predeterminer, preposition, pronoun, and verb. These three facts conspire to give a limiting factor to the total size of the grammar rule set.

This grammar is a phrase (or constituent) grammar and not a sentence grammar. The parser posits a sentence analysis only when its sentence buffer is consumed; until that point phrases are constructed and coalesced with each other as they are encountered, without regard to sentence level structure. The **Coalesce** decision is a syntactic one, with the resulting superordinate language object effectively assuming the status of the *head* of the entire existing structure. Either the new language object is inserted somewhere within the highest priority language object as a subordinate, or the highest priority object is subordinated to the new object (Fig. 4). In the second case the new object is effectively elevated to the status of superordinate, and it is this coalesced language object that is inserted into the priority queue (with a suitably computed preference score).

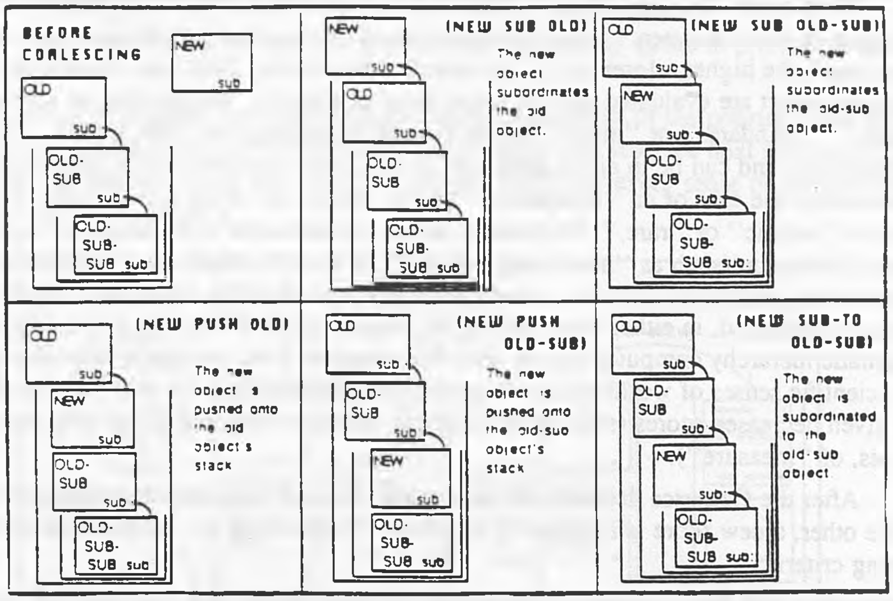
BEFORE COALESCING
NEW
sub
OLD
sub
OLD-SUB
sub
OLD-SUB-SUB sub
(NEW SUB OLD)
The new object subordinates the old object.
(NEW SUB OLD-SUB)
The new object subordinates the old-sub object.
(NEW PUSH OLD)
The new object is pushed onto the old object's stack
(NEW PUSH OLD-SUB)
The new object is pushed onto the old-sub object's stack
(NEW SUB-TO OLD-SUB)
The new object is subordinated to the old-sub object

Fig. 4: PREMO *Coalesce Operations*

At any given point in a parse, there will always be a language object construed as the *head* of the parse, whichever language object has the superordinate status, of the pair being coalesced, will become the head as per the rules of the grammar (where status is generally a reflection of the usual syntactic dominance, with PP's dominating NP's, and VP's dominating everything).

3. Preference Semantics

PREMO is a knowledge-based Preference Semantics parser (Wilks 1972, 1975a, 1975b, 1978), with access to the large, lexical semantic knowledge base created by the PREMO lexicon-provider subsystem. Preference Semantics is a theory of language in which the meaning for a text is represented by a complex semantic structure that is built up out of smaller semantic components; this compositionality is a fairly typical feature of semantic theories. The principal difference between Preference Semantics and other semantic theories is in the explicit and computational accounting of ambiguous, metaphorical, and non-standard language use.

The links between the components of the semantic structures are created on the basis of semantic preference and coherence. In text and discourse theory, coherence is generally taken to refer to the meaningfulness of text. Fass (1987) suggests that in NLP work such as Preference Semantics the notions of "satisfaction" and "violation" (of selection restrictions or preferences) and the notion of "semantic distance" (across structured type hierarchies) are different ways of characterising the meaningfulness of text; they capture different coherence relations. The original systems of Preference Semantics (Wilks 1972, 1975a, 1975b, 1978), were principally based on the coherence relation of "inclusion" (semantic preferences and selection restrictions); the emphasis in PREMO is more on the coherence relation based on semantic distance, although the original notions of coherence also survive.

In Preference Semantics the semantic representation computed for a text is the one having the most semantically *dense* structure among the competing "readings." Semantic density is a property of structures that have preferences regarding their own constituents, and satisfied preferences create density. Density is compared in terms of the existence of preference-matching features, the lack of preference-breaking features, and the length of the inference chains needed to justify each sense selection and constituent attachment decision. The job of a Preference Semantics parser, then, is to consider the various competing interpretations, of which there may be many, and to choose among them by finding the one that is the most semantically dense, and hence preferred.

3.1. Computing Preferences

When a new language object is first created it receives a preliminary preference score. An initial value is given between 1.0 and 2.0 that depends on the word's sense number (the lower the word sense, the higher, closer to 2.0, the score). Immediately thereafter, various attributes of the language object are evaluated and the initial score is adjusted. Adjustments to scores are either "minor," "standard," or "major" (in the current implementation these are 2%, 10%, and 50% respectively), and can be in either direction. In the current implementation preliminary scores are *decreased* in the case of an "interjection" part of speech, or for an LDOCE time-and-frequency code of "archaic" or "rare." Preliminary scores are *increased* if the language object is for a phrasal definition (such as "alternating current"), or is for a closed class word, 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 preferential increases and the other senses of those words are given decreased scores (such as the scientific senses, as opposed to the political and musical senses, of "measure").

After the Coalesce decision has been made, and one language object has been subordinated to the other, a new score is assigned to the result. These scores are computed according to the following criteria:

Predictions and Requirements: The lexical semantic frames have GRAMMAR slots that contain predictions and requirements: some general and some specific. Some general codes mark nouns as "a countable noun followed by the infinitive with *to*," and others mark verbs as "ditransitive and followed by a *that* clause."[1] There are also specific predictions: particular senses of "sat" and "lay" predict an adverb or preposition, and particularly "down." And there are some absolute requirements: one sense of "earth" requires the article "the." These are collected and checked as language objects are being coalesced, and subordinate predictions are "synthesized" into their superordinate language object. Naturally, when a prediction is fulfilled the preference/priority is increased; and when a prediction is made but not fulfilled, scores are decreased. The degree to which scores are changed is still being experimented with, the currently implemented heuristic is to effect larger changes for more specific predictions.

Subordination: When language objects are coalesced the current implementation awards minor increases to pairings that follow a notion of natural order; for example, a Verb phrase subordinating a Noun phrase is a natural event to expect, but an Adjective phrase subordinating a Verb phrase less so. Both must be permitted, since it is possible to construct an example of either.

Semantic Matching: Content words in the text lexicon have semantic codes placing them in the LDOCE type hierarchy (types like **abstract**, **concrete**, or **animate**). Nouns and adjectives have a single code identifying their place in the hierarchy; verbs have 1, 2, or 3 codes identifying selection restrictions on their arguments. Semantic matching is done, and scores are adjusted for semantic coherence, whenever a pair of language objects are coalesced such that (1) an adjective (or a nominal) modifies a noun phrase head, or (2) a noun phrase is being attached as an argument to a verb, or (3) a noun phrase is being attached as the object of a preposition, or (4) a prepositional phrase is being attached as an argument to a verb. In the current implementation increases (but not decreases) are computed as a function of distance in the type hierarchy. If a head prefers, say, an **animate** argument and is presented with an **abstract** word sense, the increase will be quite small as opposed to being presented with a competing **human** word sense.

3.2. Semantic Structures

When the Coalesce decision is being made, PREMO looks to see if one language object is being subordinated to the other with a "push" operation. If so, a new constituent is being started and it is appropriate to affix a semantic label onto the subordinate object, since it is about to

[1] Such as *attempt* in "an attempt to climb the mountain," and *warn* in "He warned her (that) he would come."

disappear under the new top-of-stack. These labels identify the functional or semantic relations that hold between a language object and its superordinate. The LDOCE hierarchies, in conjunction with the lexical semantic frame contained within each language object, and the "frame enriching" procedures developed for the lexicon are brought to bear at this point (Slator and Wilks, 1987, 1989); as well as the hand-coded definitions for prepositions that we must admit to creating since LDOCE, from our point of view, does not include useful case information in their preposition definitions.

Every sentence in a text is eventually represented by a single language object. These language objects are named for the word seen to be the head of the dominating phrase in the sentence, and are of type *phrase* (and presumably of phrase type VP, in the usual grammatical case). Each of the subordinated phrases in the sentence is stacked within this superordinate language object, along with a corresponding relation label.

4. PREMO Example

Consider the following sentence:

(1) *The technician measures alternating current with an ammeter.*

First PREMO loads the lexicon specific to this text, which contains 26 frames for content words. These 26 frames are: *alternate* (3 adjective senses, 1 verb sense), *ammeter* (1 noun sense), *current* (3 adjectives, 4 nouns), *measure* (8 nouns, 3 verbs, 1 adjective), *technician* (1 noun sense), and the phrase "alternating current" (1 noun sense). LDOCE defines about 7,000 phrases. PREMO performs a contextual analysis by appeal to the pragmatic codes as organized into a specially restructured hierarchy. This results in the various Science and Engineering word senses receiving increased preference/priority scores while most other word senses receive decreased scores. This context setting mechanism is discussed at length in Slator (1988a, 1988b), Slator and Wilks (1987, 1989) and Fowler and Slator (1989).

Then, PREMO initializes the priority queue (PQ) with language objects for both the adverbial and definite determiner senses of *The* (the first word in the sentence), at which point the loop of the algorithm in section 1.2, *Preference Machine Control* is entered. In the first iteration the determiner is instantiated as an Adjective phrase [AP:*The*], as is the adverb reading, according to the rules of the grammar. The analysis of sentence (1) requires a total of 12 iterations through this loop. Notice that sentence (1) is 336-way ambiguous if just the number of senses of polysemous content words are multiplied out (*alternate*=4, times *current*=7, times *measure*=12, equals 336), and that that number grows to 1008 if the three cases of *with* are included.

Iteration 2: The PQ contains three language objects for *The*, of which the definite determiner is slightly preferred. This object is popped from the queue, and the next word in its sentence buffer, *technician* is retrieved from the lexicon and instantiated as a language object. There is only a single sense of *technician* and only a single grammar rule action for the situation:

1. (AP nil noun) => (NP sub old).

This means *technician* becomes an NP with the determiner *The* subordinated to it [NP:*technician,The*]. This act of coalescing results in a minor increase being assigned to [NP:*technician,The*], and it returns to the priority queue.

Iteration 3: The language object [NP:*technician,The*] is popped from the queue and the 11 senses of the next word in its sentence buffer, *measure*, are instantiated; then 11 copies of [NP:*technician,The*] are created and paired with them. The two major grammar competitors during this iteration are:

2. (NP AP noun) => (NP sub old)
3. (NP AP verb) => (VP sub old)

that is, *measures* as a noun becoming the new head of [NP:*technician,The*], as opposed to *measures* as the head of a verb phrase taking [NP:*technician,The*] as an argument.

The criteria for comparing these readings (and the fact that PREMO prefers the second, VP analysis), reduces to the fact that the noun *measures* carries an **abstract** semantic code which does not match well with the **human** semantic code carried by *technician;* while the verb *measures* carries a **human** selection restriction for its first argument, which matches exactly with the semantic code carried by *technician*.

Iteration 4: The language object [VP:*measures*] with its subordinated language object [NP:*technician,The*] on the top of its stack, is popped from the queue and the 4 senses of *alternate*, along with the language object for the phrasal *alternating current*, are instantiated; 5 copies of [VP:*measures*] are then created and paired with them. Phrasal objects are preferred by PREMO, and nothing occurs to outweigh that preference. The question to be decided is which of these two readings should be preferred:

4. (VP NP noun) => ((NP push old) <or> (NP sub old-sub))

that is, does *alternating current* represent the start of a new NP constituent, or is it the new head of the ongoing top-of-stack NP constituent (in this case [NP:*technician,The*]). The semantic code carried by *alternating current* is **abstract-physical-quality** which is a poor match with the **human** of [NP:*technician,The*] but a good match with the second argument code of [VP:*measure*], which is **abstract.** Therefore the new NP constituent reading receives the better preference/priority score and assumes the position at the head of the PQ. However, first a label must be attached to the NP constituent that is about to disappear from the top-of-stack as a result of the "push" operation. In this case, since the verb prefers and receives a **human** subject, this NP is labelled "Agentive."

Iterations 5-11: The high priority language object on the front of the PQ is [VP:*measures*] which now subordinates both [NP:*technician,The*] and [NP:*alternating current*]. The continuation of the sentence buffer in [VP:*measures*] is now *with an ammeter*. The next several iterations are concerned with pushing a PP onto the subordinate stack of [VP:*measures*] and then subordinating [NP:*ammeter,an*] as the object of the PP. The current implementation recognizes 3 senses of the preposition *with* representing the ACCOMPANIMENT, POSSESSION, and INSTRUMENT cases. Each of these is coalesced with [NP:*ammeter,an*] and pushed onto the subordinate stack of [VP:*measures*]. The INSTRUMENT reading is preferred on the basis of semantic matching between the selection restriction code on the object of the preposition, which is **concrete,** and the semantic code for *ammeter*, which is **movable-solid.**

Iteration 12: The final iteration retrieves the language object for the [VP:*measures*] from the front of the PQ and finds the sentence buffer is empty. It is at this point that the case label marking the top-of-stack element (which is [PP:*with*,[NP:*ammeter,an*]]), as the INSTRUMENT case is actually affixed. This language object is then saved as the interpretation of sentence (1), the queue is flushed and PREMO reads whatever sentence text follows, or if none follows, PREMO ends.

4.1. Toward Solving a Hard Problem

Two contrasting methods of word sense selection have been described here. The earlier method was first explored by the Cambridge Language Research Unit, beginning in the mid-1950s. This method performed a global analysis, (Masterman 1957, as described in Wilks, 1972), that relied on a thesaural resource. This method was good at choosing word senses coherent with the other words in the text, by using a system of looking to the sets of words found under common thesaural heads, and performing set intersections. The problem is that the less "coherent" word senses are missed and so, for example, in a physics text only the scientific sense of *mass* will chosen and, therefore, the word *mass* in the phrase *mass of data* will come out wrong in that text.

The other method is Preference Semantics that performs a local analysis that relies on semantic type markers. This method is good at choosing word senses that best fit with other words in a sentence, by using a system of matching and comparing among the various primitive elements that make up the meanings of words. The problem is that this can be fooled into preferring word senses that only seem to be best. The standard example, attributed to Phil Hayes, is the following.

(2) *A hunter licked his gun all over and the stock tasted good.*

In this example, the challenge is to choose the correct sense of *stock*. The problem is that the local evidence supplied by the verb *to taste* points towards the "stock as soup" reading, which is wrong. Granted, this is something of a pathological example, but it is famous and it captures the flavor of the objection.

PREMO attempts to tackle these contrasting analysis problems by bringing together both global and local information. To demonstrate this, consider the analysis of the following two short texts. The first is familiar from the example in Section 4, above.

(3) *Current can be measured.*
The technician measures alternating current with an ammeter.

The following text is intended to parallel the first one.

(4) *Current can be measured.*
The geographer measures river basin flow near a lake.

The point at issue is choosing the correct sense of *current* in each case. In text (3) it is the **engineering/electrical** sense that should be chosen. In text (4), however, it is the **geology-and-geography** sense of *current* that is correct. In the absence of other evidence, the **geology-and-geography** sense is the one most systems would choose, since this is the more common usage of the word in the language (and the lowest numbered word sense in LDOCE, which reflects this notion of default preference). And since most systems have no notion of global text coherence, most would get the wrong sense of *current* in text (3) at first reading. It is conceivable that the sense selection for *current* could be corrected after further text has been processed, but few if any systems attempt this, and it is far from obvious how this should be done in general. PREMO gets both of these texts right, by choosing the correct sense of *current* in each case, and making all of the other word sense and attachment decisions (see fig. 5).

In Fig. 5, each line element represents a condensation of a language object. Language objects are complex items which this diplay merely summarizes. The form of these displays is as follows:

(<name> <phrase-type> <score> <part-of-speech> <sense-number> <stack-labels>)

and the lower language objects are on stacks, as indicated by their indentations. And so, the first language object reads as this: the third sense of the verb *measure* (the linking verb sense), has two elements on its subordinate stack, one marked as a verb inflection (the language object for *be*, which itself has a language object on its internal stack marking verb inflection, the language object for *can*), and the other stack element (note, the second sense of *current*), marked as the subject of the measuring. The third language object in Fig. 5 has the identical interpretation, except that the subject of the measuring is the first sense of *current* rather than the second.

PREMO analysis for text (3) *Current can be measured.*

```
((measured VP 4.855569S0 "v" "0300" (INFL SUBJECT))
 (((be VP 4.197S0 "v" "0008" (INFL))
   ((can VP 3.174081S0 "v" "0100" nil)))
  (Current NP 2.985294S0 "n" "0200" nil)))
```

PREMO analysis for *The technician measures alternating current with an ammeter.*

```
((measures VP 4.68685S0 "v" "0100" (INSTRUMENT OBJECT1 AGENTIVE))
 (((with PP 3.726117S0 "prep" "0000" (OBJECT1))
   (((ammeter NP 2.814706S0 "n" "0000" (DET))
     ((an AP 1.815S0 "indef" "0000" nil)))))
  (alternating*current NP 3.838235S0 "n" "0000" nil)
  ((technician NP 0.8368717S0 "n" "0000" (DET))
   ((The AP 0.9982499S0 "definite" "0100" nil)))))
```

PREMO analysis for text (4) *Current can be measured.*

```
((measured VP 4.757666S0 "v" "0300" (INFL SUBJECT))
 (((be VP 4.077573S0 "v" "0008" (INFL))
   ((can VP 3.028026S0 "v" "0100" nil)))
  (Current NP 2.763158S0 "n" "0100" nil)))
```

PREMO analysis for *The geographer measures river basin flow near a lake.*

```
((measures VP 4.354593S0 "v" "0100" (LOCATIVE OBJECT1 AGENTIVE))
 (((near PP 2.406232S0 "prep" "0000" (OBJECT1))
   (((lake NP 2.719298S0 "n" "0000" (DET))
     ((a AP 0.9075S0 "indefinite" "0100" nil)))))
  ((flow NP 1.433986S0 "n" "0500" (KIND-OF))
   (((river*basin NP 2.178159S0 "n" "0000" (DET))
     ((the AP 0.9982499S0 "definite" "0100" nil)))))
  ((geographer NP 1.693873S0 "n" "0000" (DET))
   ((The AP 0.9982499S0 "definite" "0100" nil)))))
```

Fig. 5: PREMO Analyses for Texts (3) and (4).

The other two language objects displayed both show the first sense of *measure* each with three language objects on the subordinate stack. In each case these subordinate language object constituents represent the AGENT (*technician* and *geographer*), and the OBJECT (*alternating current* and *river basin flow*), of the measuring action. Text (3) also has a prepositional phrase attached in the INSTRUMENT case while text (4) has a prepositional phrase attached in the LOCATIVE case.

In spite of this success, the problem of mediating the tension between global and local sources of information is still not completely solved. PREMO assumes text coherence and so while Preference Semantics provides a naturally local sort of analysis procedure these local effects can be overcome by appeal to global context. However, it is still possible to find examples, such as text (2) above, that do not *have* any context (a common situation in the computational linguistics literature, where space constraints and custom preclude long examples). And in the absence of this global information PREMO will not perform any better than any other system of analysis. That is, if text (2) were embedded in a longer exposition about hunters and guns, then the probability is high that the correct sense of *stock* would be chosen. If however, text (2) were embedded in an exposition about food and cooking, PREMO would almost certainly get this wrong. And in the absence of context PREMO will choose the ''soup stock'' reading because the notion of gun stocks having a taste is not one that finds much support in a system of semantic analysis.

5. Comparison to Other Work

The original Preference Semantics implementations (Wilks 1972, 1975a, 1975b, 1978), operated over a hand coded lexicon of semantic formulae. Input strings were segmented into phrases beforehand, and coherence was essentially a matter of counting ''semantic ties'' inferred by pattern matching between formulae. PREMO operates over a machine-readable lexicon that is at once much broader and shallower than the original. To make up for this, preference scoring in PREMO is much more finely grained and takes more into account (grammatical predictions, pragmatic context, etc.). If anything, PREMO is grammatically weaker than the original work, while being more robust in the sense that syntactic anomalies and ill-formed input are processed the same as anything else.

A group at Martin Marietta (Johnson, Kim, Sekine, and White, 1988; White 1988), built a language understander based on Preference Semantics, but modified by their own interpretation of Wilks, Huang, and Fass (1985). Their NLI system is frame-based and much of the system's knowledge resides in the lexicon, which is constructed by hand. The parsing process is separated into three autonomous modules: BuildRep (a constituent parser), Validate (a sort of constituent filter), and Unify (which incrementally builds a semantic structure from validated constituents). The principle differences between their system and PREMO lies in their criteria for abandoning non-productive paths (their domain constraints allow them to prune on the basis of semantic implausibility), and in their lack of a high level control structure (it is possible in there system for every parse to be abandoned and nothing returned).

Preference Semantics has also been used for parsing by Boguraev (1979), Carter (1984, 1987), and Huang (1984, 1988). However, this work uses a conventional parsing strategy in which syntax drives the parsing process depth-first and Preference Semantics is used within a semantics component that provides semantic verification of syntactic constituents. PREMO has a more flexible, more breadth-first parsing strategy in which syntax, semantics, and pragmatics interact more freely. The Meta5 semantic analyzer of Fass (1986, 1987, 1988), based on the system of Collative Semantics, which extends Preference Semantics, operates over a rich hand-coded lexicon comprised of a network of "sense frames." The principal goal of this system is to identify and resolve metaphorical and metonymous relations and, with its rich semantic knowledge base, Meta5 is able to produce deep semantic analyses which are quite impressive, although constrained to a somewhat narrow range of examples.

6. Discussion

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. PREMO intentionally exploits everything that LDOCE offers, particularly in the area of grammatical predictions, and also in terms of the TYPE hierarchy as given, and the PRAGMATIC hierarchy as restructured, as well as extracting semantic information from the text of definitions.

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). Another design principle is to cut every conceivable corner by making "smart" preference evaluations. The potential remains however, for worst case performance, where the preference/priority scores work out so that every newly coalesced pair immediately gets shoved to the bottom of the priority queue. If this happens the algorithm reduces to a brute search of the entire problem space.

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 intuitive plausibility. The competition between parses, based as it is on the tension between the various preference/priority criteria is vaguely reminiscent of a "spreading activation" system where the various interpretations "fight it out" for prominence. The PREMO architecture is, of course, utterly different in implementation detail, and it is not at all obvious how it could be equivalently converted, or that this metaphor is even a fruitful one. 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 parallel processor (like a Hypercube). Each of the pairs would be distributed out to the (cube) processing elements where the coalescing and preference/priority scoring would be done in parallel.

References

BOGURAEV, BRANIMIR K. (1979). Automatic Resolution of Linguistic Ambiguities. *University of Cambridge Computer Laboratory Technical Report*. (No.11). Cambridge, UK: University of Cambridge Computer Laboratory.

CARTER, DAVID M. (1984). An Approach to General Machine Translation Based on Preference Semantics and Local Focussing. *Proceedings of the 6th European Conference on AI (ECAI-84)*, pp. 231-238. Pisa, Italy.

CARTER, DAVID M. (1987). *Interpreting Anaphors in Natural Language Texts*. Chichester, UK: Ellis Horwood.

FASS, DAN C. (1986). Collative Semantics: An Approach to Coherence. *Computing Research Laboratory Memorandum*. (MCCS-86-56). Las Cruces, NM: New Mexico State University.

FASS, DAN C. (1987). Semantic Relations, Metonymy, and Lexical Ambiguity Resolution : A Coherence-Based Account. *Proceedings of the 9th Annual Cognitive Science Society Conference*, pp. 575-586. Seattle, WA: University of Washington.

FASS, DAN C. (1988). An Account of Coherence, Semantic Relations, Metonymy, and Lexical Ambiguity Resolution. In *Lexical Ambiguity Resolution in the Comprehension of Human Language*. Edited by Steve L. Small, Gary W. Cottrell, and Michael K. Tanenhaus. pp. 151-178. Los Altos, CA: Morgan Kaufmann.

FOWLER, RICHARD H. AND BRIAN M. SLATOR (1989). Information Retrieval and Natural Language Analysis. *Proceedings of the 4th Annual Rocky Mountain Conference on Artificial Intelligence (RMCAI-89)*, pp. 129-136. Denver, CO. June 8-9

HUANG, XIUMING (1984). A Computational Treatment of Gapping, Right Node Raising and Reduced Conjunction. *Proceedings of the 10th International Conference on Computational Linguistics (COLING-84)*, pp. 243-246. Stanford, CA.

HUANG, XIUMING (1988). XTRA: The Design and Implementation of A Fully Automatic Machine Translation System. *Computing Research Laboratory Memorandum*. (MCCS-88-121). Las Cruces, NM: New Mexico State University.

JOHNSON, HEIDI, G. KIM, Y. SEKINE, AND JOHN S. WHITE (1988). Application of Natural Language Interface to a Machine Translation Problem. *Proceedings of the Second International Conference on Theoretical and Methodological Issues in Machine Translation*. Pittsburgh, PA. June.

MASTERMAN, M. (1957). The Thesaurus in Syntax and Semantics. *Mechanical Translation*, 4, 1-2.

PROCTER, PAUL ET AL. (1978). *Longman Dictionary of Contemporary English (LDOCE)*. Harlow, Essex, UK: Longman Group Limited.

SLATOR, BRIAN M. (1988a). Constructing Contextually Organized Lexical Semantic Knowledge-Bases. *Proceedings of the Third Annual Rocky Mountain Conference on Artificial Intelligence*, pp. 142-148. Denver, CO. June, 13-15.

SLATOR, BRIAN M. (1988b). Lexical Semantics and a Preference Semantics Analysis. *Memoranda in Computer and Cognitive Science*. (MCCS-88-143). Las Cruces, NM: Computing Research Laboratory, New Mexico State University. (Doctoral Dissertation).

SLATOR, BRIAN M. (1988c). PREMO: the PREference Machine Organization. *Proceedings of the Third Annual Rocky Mountain Conference on Artificial Intelligence*, pp. 258-265. Denver, CO. June, 13-15.

SLATOR, BRIAN M. AND YORICK A. WILKS (1987). Towards Semantic Structures from Dictionary Entries. *Proceedings of the Second Annual Rocky Mountain Conference on Artificial Intelligence*, pp. 85-96. Boulder, CO. Also as CRL Memo MCCS-87-96.

SLATOR, BRIAN M. AND YORICK A. WILKS (Forthcoming - 1989). Towards Semantic Structures from Dictionary Entries. In *Linguistic Approaches to Artificial Intelligence*. Edited by Andreas Kunz and Ulrich Schmitz. Frankfurt: Peter Lang Publishing House. Revision of RMCAI-87 and CRL-MCCS-87-96.

WHITE, JOHN S. (1988). Advantages of Modularity in Natural Language Interface. *Proceedings of the Third Annual Rocky Mountain Conference on Artificial Intelligence*, pp. 248-257. Denver, CO. June, 13-15.

WILKS, YORICK A. (1972). *Grammar, Meaning, and the Machine Analysis of Language*. London: Routledge and Kegan Paul.

WILKS, YORICK A. (1975a). An Intelligent Analyzer and Understander of English. *Communications of the ACM*, 18, 5, pp. 264-274. Reprinted in "Readings in Natural Language Processing," Edited by Barbara J. Grosz, Karen Sparck-Jones and Bonnie Lynn Webber, Los Altos: Morgan Kaufmann, 1986, pp. 193-203.

WILKS, YORICK A. (1975b). A Preferential Pattern-Seeking Semantics for Natural Language Inference. *Artificial Intelligence*, 6, pp. 53-74.

WILKS, YORICK A. (1978). Making Preferences More Active. *Artificial Intelligence*, 11, pp. 75-97.

WILKS, YORICK A., DAN C. FASS, CHENG-MING GUO, JAMES E. MCDONALD, TONY PLATE, AND BRIAN M. SLATOR (1987). A Tractable Machine Dictionary as a Resource for Computational Semantics. *Proceedings of the Workshop on Natural Language Technology Planning*, Sept. 20-23. Blue Mountain Lake, NY. Also as CRL Memo MCCS-87-105. To appear in "Computational Lexicography for Natural Language Processing." Branimir K. Boguraev and Ted Briscoe (Eds.). Harlow, Essex, UK: Longman. 1988

WILKS, YORICK A., DAN C. FASS, CHENG-MING GUO, JAMES E. MCDONALD, TONY PLATE, AND BRIAN M. SLATOR (1988). Machine Tractable Dictionaries as Tools and Resources for Natural Language Processing. *Proceedings of the 12th International Conference on Computational Linguistics (COLING-88)*. Budapest, Hungary. Aug. 22-27.

WILKS, YORICK A., DAN C. FASS, CHENG-MING GUO, JAMES E. MCDONALD, TONY PLATE, AND BRIAN M. SLATOR (Forthcoming - 1989). Providing Machine Tractable Dictionary Tools. In *Theoretical and Computational Issues in Lexical Semantics*. Edited by James Pustejovsky. Cambridge, MA: MIT Press.

WILKS, YORICK A., XUIMING HUANG, AND DAN C. FASS (1985). Syntax, Preference, and Right Attachment. *Proceedings of IJCAI-85*, 2, pp. 779-784. Los Angeles, CA.