

Preference Semantics for Message Understanding

Ralph Grishman and John Sterling
Computer Science Department
New York University

1. The Problem: Capturing the meaningful semantic relations

The design of effective natural language processing systems requires a combination of the theoretical and the practical. We want to have a theoretically well-founded design so that we can take advantage of gradual improvements in our knowledge of syntax, semantics, discourse structures, and the subject domain. At the same time we need to adopt a practical approach which recognizes the inevitable shortcomings of our knowledge in these areas. We need to create *robust* systems which are able to deal appropriately with these shortcomings. We are interested in particular in systems for extracting specified information from a text. Such systems are robust if they are able to extract at least partial information despite the presence of ill-formed or unexpected syntactic, semantic, or discourse structures.

One type of knowledge which is central to most language understanding systems, in one form or another, is knowledge of the set of meaningful semantic relations in a domain. For most realistic domains, however, this set is very large and not strictly closed. Accumulating a complete inventory of these relations is therefore very difficult, if not impossible. Practical language analysis systems must be able instead to operate with an incomplete knowledge of these relations.

2. The Approach: Using Preference Semantics

This knowledge of semantic relations is typically encoded as a set of semantic patterns or "models". In our system, the entities and predicates of the domain (in linguistic terms, the nouns, verbs, and adjectives) are first grouped into semantic classes, forming a classification hierarchy. We then provide for each noun, verb, and predicate adjective (either individually or as part of a semantic class) a semantic model which describes the meaningful operands (subjects, complements, and modifiers) of that word. For each operand we specify its semantic class and its position (subject, direct object, indirect object) or syntactic marker (governing prepositions). We also mark each operand as *required* (must appear explicitly in the text), *essential* (may be omitted in the text but is required in the logical representation; semantic analysis will attempt to recover the operand), or *optional* (not required in either text or logical representation).¹ If a word has several senses, it will in general have several models.

These models serve two primary functions: as selectional constraints, to select the correct parse, and for mapping the linguistic structures into a set of domain predicates which are used for further processing. During the parse, whenever a noun phrase or clause is completed, it is matched against the semantic models. In our prior systems, selectional constraints were strictly enforced: if the phrase or clause did not match the model, it was rejected. After the parse is completed, the models are used for creating a semantic representation: within verb models, we indicate the corresponding domain predicate and, for each operand, the corresponding argument of that predicate.

To enforce selectional constraints in the context of incomplete semantic knowledge, we use a scheme of *preference semantics*, as introduced by Wilks (1975): we seek the parse violating the

¹ For a discussion of essential roles, see (Palmer 1986).

fewest selectional constraints. More precisely, if a clause or noun phrase has all the arguments required by a semantic model, but also has some arguments or modifiers not allowed by the model, we associate a penalty with each extraneous argument/modifier. If a clause or noun phrase does not match any model, we assign a larger penalty. These penalties are added together to get a score for the entire parse. We use a best-first search to find the parse with the lowest penalty.

These partial matches are also the basis for mapping the syntactic analysis into domain predicates/arguments. Extraneous operands, and clauses and noun phrases not matching any semantic model, are ignored in the mapping process. We insist that any *required* arguments be present, so that the argument structure of the created predicate will be complete. As a result, the stages which operate on the semantic structures (simplification, anaphora resolution, discourse analysis) need not be aware of the use of preference semantics.

In many applications, it is possible to identify a set of critical predicates, representing the most important events or assertions in a class of texts. An analysis of a text will be deemed successful (for such an application) if we are able to identify the instances of these predicates, along with their arguments, in the text. We therefore include among the semantic models these predicates, in their various linguistic realizations, along with "higher-order" (modal, epistemic, etc.) verbs and other frequent verbs. Preference semantics will then tend to guide the parser towards correct analyses of clauses involving these critical predicates, even if modifiers which are not modeled end up being incorrectly analyzed and/or ignored.

3. Application and Evaluation

This approach has been implemented as part of the PROTEUS message understanding system, and evaluated as part of the recent MUCK-II conference (Sundheim 1989a). This conference involved the development of a system for processing Navy OPREP (OPERational REPort) messages; these messages are reports of sightings and engagements at sea (Sundheim 1989b). They include brief (typically 3 or 4 "sentence") narrative sections, most often in a telegraphic style with omitted subjects and objects and run-on sentences. The task of the participants in the conference was to extract specified information from the narratives (ignoring the other fields of the message). Our systems had to identify five types of events in the message (detecting, tracking, harassing, targeting, and attacking) and in each case whether the initiating force was friend or foe. For each such event the system was to create a frame and fill in 8 additional slots concerning the agent, instrument, time, etc.

The principal stages of the PROTEUS message understanding system are syntactic analysis, semantic analysis (translation to predicate form), anaphora resolution, discourse analysis, and frame (data base) creation. In order to analyze the telegraphic text, the English grammar is extended to include a variety of fragmentary constructions, but with a penalty so that full sentence analyses are preferred (Grishman 1989). Selectional constraints are imposed (penalties are computed) during parsing whenever a noun phrase or clause is completed.

As part of the MUCK evaluation, a total of 125 messages were provided over a period of 3 months prior to the evaluation (105 initially, 20 more a month before the conference). A final set of 5 messages were used for an on-site evaluation at the conference. Time constraints prevented us from including semantic models for many of the constructs present in the messages. We instead relied heavily on preference semantics in order to "get through" the analysis of many of the sentences. Since the task specified the types of events of interest, we could begin by creating semantic models for the corresponding verbs and nominalizations (for detection, attack, etc.), along with noun phrase models for the classes of arguments (missiles, ships, planes, etc.). From this starting point we gradually extended the model coverage to include higher order verbs and a few other frequent verbs from the messages.

Quite often (although by no means always) the result of preference semantics was to get a correct analysis with one argument or modifier ignored. For example, for the sentence

FRIENDLY B-52 ON MINING MISSION ESCORTED BY AMERICA F-14'S WERE ATTACKED BY FOUR HOSTILE MIG-21'S AND ONE BISON.

the model for *plane* does not include any modifiers such as ON MINING MISSION, so this phrase is ignored; this does not affect the data base entry generated. For the sentence

S-3'S LAUNCHED 4 SAMSON FOLL BY EA-6 HARM.

we have no model for "followed by *missile*", so the phrase FOLL BY EA-6 HARM is ignored; in this example we lose one of the agents and instruments of the attack.

Some statistics on the main training corpus of 105 messages will give some indication of the significance of preference semantics in processing these texts. Of the total of 305 "sentences" (sequences ending in periods or field terminators), we obtained a syntactic analysis (not necessarily completely correct) for 288.² Preference semantics was used for 116 of these (i.e., the analyses of these sentences had one or more phrases not matching the semantic model). In terms of task performance: NOSC determined that the 105 messages should have generated a total of 132 entries in the "event" data base.³ Using preference semantics, our system correctly identified (in terms of level of action and initiating force) 101 (77%) of these;⁴ without preference semantics, it was only able to correctly identify 74 (56%).

4. Discussion

The effect of removing preference semantics would have been greater were it not for the presence of other mechanisms included in our system to enhance robustness. One of these is the arrangement of the semantic models in a hierarchy, so that if a model for a specific noun or verb fails to match, an attempt will be made to match a more general model. Another is a 'longest parse' mechanism which, if no analysis can be obtained for the entire sentence, takes the longest substring, starting with the first word, for which an analysis was obtained.

We may expect that as one robustness mechanism is removed, others will play a larger role. We can see this effect between preference semantics and the longest parse mechanism. When running with preference semantics, the system resorts to the longest parse heuristic 42 times (246 other sentences got full parses); when preference semantics is disabled (i.e., selection is strictly enforced), the system used longest parse 83 times (68 others got full parses). This effect can be understood as follows: if the sentence contains a modifier which does not fit the semantic model, preference semantics will incorporate it into the sentence analysis with a penalty. If preference semantics is disabled and the modifier is near the end of the sentence, we may be able to obtain an analysis of the text up to the beginning of the modifier as a complete sentence or sentence fragment; this analysis will be returned by the longest parse heuristic.

If *both* preference semantics and the longest parse mechanism are disabled, we are left with only 68 sentences which can be analyzed. The task performance plummets accordingly: only 43 (33%) of the events are correctly identified. These results can be summarized in a table:

² For 246 sentences, we obtained a parse of the entire sentence; for an additional 42, a parse of a substring of the sentence. See section 4 for further discussion.

³ In addition, 6 messages had no events of these 5 types, and generated "OTHER" entries in the data base. They are not included in the counts given here.

⁴ The system as run at the MUCK-II conference, and as reported on at the DARPA workshop, correctly identified 99 events. However, in preparing this paper we have found and corrected a small error in the selection mechanism, and rerun all the experiments with this correction. This has resulted in small changes in some of the figures reported.

heuristics used	full sent. parses	substring parses	# (%) of events identified
preference semantics and longest parse	246	42	101 (77%)
longest parse	68	83	74 (56%)
neither	68	0	43 (33%)

The specific numbers presented here are not especially significant, since they reflect the incompleteness of the semantic model at the time of our evaluation. Our semantic model was constructed entirely by hand; for future evaluations, we hope that larger text samples coupled with more automated procedures for model acquisition (as described, for example, in (Grishman 1986) and (Lang 1988)) will allow us to provide broader model coverage within similar time constraints. However, even with the best tools significant gaps will be unavoidable in a model for a large domain. This paper has indicated how, under these circumstances, relatively simple mechanisms can be used to boost the performance of text understanding systems.

Acknowledgement. This research was supported by the Defense Advanced Research Projects Agency under contract N00014-85-K-0163 from the Office of Naval Research.

References

(Grishman 1986)

Ralph Grishman, Lynette Hirschman and Ngo Thanh Nhan, Discovery Procedures for Sublanguage Selectional Patterns: Initial Experiments. *Computational Linguistics*, 12 (3) 205-216, 1986.

(Grishman 1989)

Ralph Grishman and John Sterling, Analyzing Telegraphic Messages. *Proc. Speech and Natural Language Workshop*, Philadelphia, PA, February, 1989, Morgan Kaufmann.

(Lang 1988)

Francois-Michel Lang and Lynette Hirschman, Improved Portability and Parsing through Interactive Acquisition of Semantic Information. *Proc. Second Conf. Applied Natural Language Processing*, Austin, TX, February, 1988.

(Palmer 1986)

Martha Palmer, Deborah Dahl, Rebecca Schiffman, Lynette Hirschman, Marcia Linebarger, and John Dowding, Recovering Implicit Information. *Proc. 24th Annl. Meeting Assn. Computational Linguistics*, 10-19, 1986.

(Sundheim 1989a)

Beth Sundheim, Plans for a task-oriented evaluation of natural language understanding systems. *Proc. Speech and Natural Language Workshop*, Philadelphia, PA, February, 1989, Morgan Kaufmann.

(Sundheim 1989b)

Beth Sundheim, *Navy Tactical Incident Reporting in a Highly Constrained Sublanguage: Examples and Analysis*. Naval Ocean Systems Center Technical Document 1477.

(Wilks 1975)

Yorick Wilks, An intelligent analyzer and understander of English. *Comm. Assn. Comp. Mach.* 18, 264-274, 1975.