Information Extraction and Semantic Constraints

Ralph Grishman and John Sterling

Computer Science Department
New York University
New York, NY 10003, USA
grishman@nyu.edu

Abstract

We consider the problem of extracting specified types of information from natural language text. To properly analyze the text, we wish to apply semantic (selectional) constraints whenever possible; however, we cannot expect to have semantic patterns for all the input we may encounter in real texts. We therefore use *preference semantics*: selecting the analysis which maximizes the number of semantic patterns matched. We describe a specific information extraction task, and report on the benefits of using preference semantics for this task.

Task and Approach

Information extraction is the task of extracting specified types of information from a natural language text — for example, information about specific classes of events. Typically, however, the text to be processed will contain many types of events besides the classes of interest. The system designer therefore faces a quandary in imposing semantic (selectional) constraints. Selectional constraints could be strictly enforced: a sentence analysis is not accepted unless all relationships are identified as semantically valid. In this case, the designer either must encode all the semantic relationships which may occur in the text — an impractical if not impossible task — or be resigned to losing events of interest occurring in sentences which also contain unexpected semantic relationships. On the other hand, if selectional constraints are not enforced, sentences containing events of interest may be incorrectly analyzed.

Several approaches have been suggested to extricate ourselves from this quandary. One approach has been an analyzer driven by semantic expectations, ignoring intervening text not matching these expectations [1]; this is robust but can lead to serious errors. Another approach has been to identify "interesting" words and attempt only partial sentence parses around those words [2]. As an alternative, we have explored the use of full syntactic analysis of the input, coupled with *preference semantics*. Preference semantics, as introduced by Wilks [3], penalizes but does not reject analyses which violate semantic constraints; it selects the analysis with the fewest constraint violations.

The task to which we have applied preference semantics is that of creating a data base from U S Navy messages describing naval encounters. These messages are relatively brief (average length 30 words) and are highly telegraphic, with many sentence fragments and frequent run-on sentences. The specific task was to identify five classes of events within these messages and, for each event, identify the initiating force (friend or foe) and 8 other parameters (agent, object, instrument, location, time, etc.). Our and other systems were ported to this domain and evaluated over a period of 3 months in the spring of 1989 as part of Message Understanding Conference-II [4] (held at the Naval Ocean Systems Center, San Diego, California, USA, in June 1989).

System Design

The principal system components are[1]

- a syntactic analyzer, using an augmented context-free grammar, which produces a parse and a regularized syntactic structure
- a semantic analyzer, which maps clauses and nominalizations into domain-specific predicates
- reference resolution, which determines referents for anaphoric and omitted arguments
- data base creation, which maps predicates describing events of interest into data base entries

[1] In addition to these principal components, there is a small semantic regularization component (following semantic analysis), which performs some decomposition and simplification of semantic forms. There is also a discourse analysis component (following reference resolution) which identifies possible causal and enabling relations in a message. If reference resolution generates alternative hypotheses, those leading to the identification of such relations in the message will be preferred. We found, however, that in our application discourse analysis made only a minimal contribution to overall system performance.

The telegraphic message style is accomodated explicitly in the grammar, following the approach of Marsh and Sager [5], by including productions for various fragment types in the grammar. Run-on sentences are also explicitly provided for in the grammar. Some inputs can be analyzed either as full sentences or as fragments; we prefer the full-sentence analysis by associating a penalty (reduced score) with fragment analyses and using a best-first search algorithm in the parser. The reference resolution component assists in the analysis of fragments by attempting to recover omitted but semantically essential arguments[2] (a similar approach is taken in [6]).

The verbs, nouns, and entity names are organized into a domain-specific semantic classification hierarchy. Knowledge of the meaningful semantic relationships is then encoded as a set of patterns for each noun and verb, indicating the semantic class of each argument and modifier, and whether the argument is required or optional. This knowledge plays a role at two points in the analysis process. During parsing it is used to check selectional constraints; during semantic analysis it is used to guide the mapping into domain predicates.

In keeping with the basic tenet of preference semantics, we do not require a perfect match between the input and our semantic patterns. Beyond that, however, our approach differs from Wilks', reflecting the difference in our analysis procedures (we perform a full syntactic analysis, whereas Wilks did not) and in our application. In enforcing selectional constraints, we insist that all required arguments be present. We impose a small penalty for extraneous arguments and modifiers (phrases in the input which do not match the pattern) and a larger penalty for clauses and noun phrases which do not match any pattern at all. These penalties are applied during parsing, and are combined with the syntactic penalties (for sentence fragments) noted above. We then use our best-first parser to seek the analysis with the lowest penalty. In the process of mapping into domain predicates, we ignore these extraneous arguments and modifiers.

These messages contain a wide variety of information besides the events identified as being of interest; it was not feasible to incorporate semantic patterns for all these verbs and noun phrases. Rather, we confined ourselves to creating patterns for the events and objects of interest, verbs and adjectives with sentential complements ("began to ___", "unable to ___"), and a few other high-frequency verbs. In principle, this would allow us to get correct analyses for sentences or portions of sentences containing events of interest, while preference semantics would allow us to "get through" the remaining text.

[2] Following the terminology of [6], arguments which must be present in the input text are termed *required*, while arguments which may be absent in the input text but must be present in the final logical form are termed *essential*.

Results

The effects of switching from strict selection to preference semantics were dramatic. The main training corpus contained 105 messages with 132 events to be identified. With strict selection, only 43 (33%) were correctly identified as to type of action and initiating force; with preference semantics, this improved to 90 events (68%). With further heuristics, described in [7], our system was able to correctly identify 101 (77%).

Interestingly, the number of incorrect data base entries[3] generated increased only slightly: from 10 with strict selection to 13 with preference semantics (and did not increase further with the additional heuristics), while the omission rate, of course, went down sharply. This may be a consequence of our conservative semantic interpretation strategy, which will make use of the semantics of an embedded structure only if the higher-level structure in which it is embedded has been "understood" (matched to a pattern). For example, this would avoid the extraction of the information "ship was sinking" from the phrase "denied that ship was sinking" if we did not have any semantics for "deny".

Concluding Remarks

Like others who are attempting to construct robust text analysis systems (e.g., [8]), we believe that the key lies in the successful integration of a variety of constraints: syntactic, semantic, domain, and discourse information. We want these constraints to be as rich as possible, yet we also recognize that, because of system limitations and ill-formed input, each may be violated. To allow for this, we associate a penalty with each violation and seek a 'best analysis' which minimizes these penalties. We have demonstrated the effectiveness of this

[3] Event records with an incorrect type of action or initiating force.

approach with regard to semantic constraints and a limited set of syntactic preferences (preferring whole sentence to fragment analyses). We are currently experimenting with a stochastic grammar (trained on a sample corpus of messages) in order to provide a richer and systematically derivable set of syntactic preferences.

Implementation

This system is implemented entirely in Common Lisp and has been run on a Symbolics LISP machine.

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

[1] G. F. DeJong, An Overview of the FRUMP System. In W. G. Lenhert and M. H. Ringle (eds.), *Strategies for Natural Language Processing*. Lawrence Erlbaum Assoc., Hillsdale, NJ, 1982, pages 149-176.

[2] David Allport, The TICC: parsing interesting text. *Proc. Second Conf. Applied Natural Language Processing*. 1988, pages 211-218.

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

[4] 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, pages 197-202.

[5] Elaine Marsh and Naomi Sager, Analysis and Processing of Compact Text. *Proc. COLING 82*, pages 201-206.

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

[7] R. Grishman and J. Sterling, Preference Semantics for Message Understanding. *Proc. Speech and Natural Language Workshop*, Harwich Port, MA, October, 1989, Morgan Kaufmann, pages 71-74.

[8] Lisa Rau and Paul Jacobs, Integrating Top-down and Bottom-up Strategies in a Text Processing System. *Proc. Second Conf. Applied Natural Language Processing*, 1988, pages 129-135.