Parsing Run Amok: Relation-Driven Control for Text Analysis *

Paul S. Jacobs
Artificial Intelligence Laboratory
GE Research and Development Center
Schenectady, NY 12301 USA
psjacobs@crd.ge.com

Abstract

Traditional syntactic models of parsing have been inadequate for task-driven processing of extended text, because they spend most of their time on misdirected linguistic analysis, leading to problems with both efficiency and coverage. Statistical and domain-driven processing offer compelling possibilities, but only as a complement to syntactic processing. For semantically-oriented tasks such as data extraction from text, the problem is how to combine the coverage of these "weaker" methods with the detail and accuracy of traditional lingusitic analysis. A good approach is to focus linguistic analysis on *relations* that directly impact the semantic results, detaching these relations from the complete constituents to which they belong. This approach results in a faster, more robust, and potentially more accurate parser for real text.

1 Introduction

During the last several years, the field of natural language understanding research has moved very quickly from the analysis of single sentences in interface applications to the accurate extraction of data from large bodies of real text. The rapid scale-up of text interpretation systems seems close to producing programs that can accurately extract shallow data from broad volumes of text, such as a daily newspaper or reference volume. These emerging applications demand new technologies, especially in the control of parsing.

The task of adequately processing naturally-occurring texts is so difficult for traditional models of language analysis that systems that do relatively little parsing often appear to do as well [Sundheim, 1989; Sharman *et al.*, 1990; Church *et al.*, 1989] as programs that can parse broadly using well-developed grammars and models of language. However, in task-driven evaluations such as the recent DARPA-sponsored Message Understanding Conference (MUC-3) [Sundheim, 1991; Lehnert and Sundheim, 1991], the programs that combine sound partial parsing with other strategies seem to outperform both the traditional parsing and the no-parsing approaches. The message of these experiments is that weak methods are not superior to parsing, but that parsing must be carefully controlled to produce useful information from text. Broad-coverage parsers "run amok" [1] when confronted with extended texts without sufficient information to control the interpretation process.

Relation-driven control is a partial parsing method that combines the benefits of linguistic parsing with domain-driven analysis and "weak" methods by combining information at the level of linguistic *relations*, using these relations to guide parsing, recovery, and semantic interpretation. The motivation for this polytheoretic approach is that full parsing really outperforms weak methods when there is sufficient knowledge to control parsing and to produce meaningful results when the parser fails. Linguistic relations, such as *subject-verb* and *verb-complement*, are the focal point of this method because they relate to lexical and semantic preferences better than individual lexical items or complex syntactic structures.

The relation-driven control method is implemented in the current version of the GE NLToolset [Jacobs and Rau, 1990b], although many aspects of the method are incomplete. In this preliminary form, the approach tested successfully in the government-sponsored MUC-3 evaluation [Krupka *et al.*, 1991b], showing significant if not dramatic effects on system coverage and speed without a loss of accuracy. The rest of this paper will give some examples of the problems with full text parsing, present the relation-driven control strategy with results from processing representative samples of text, and ex-

*This research was sponsored (in part) by the Defense Advanced Research Project Agency (DOD) and other government agencies. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Project Agency or the US Government.

[1] **amok** (Webster's New World Dictionary, 1988): Malay *amuk*, attacking furiously...**run** (or **go**) **amok** **1** to rush about in a frenzy to kill **2** to lose control of oneself and behave outrageously or violently **3** to become wild or undisciplined

plain how the method impacts the performance of the text extraction system.

2 Problems in extracting data from real text

No data extraction system can adequately process an arbitrary text in an arbitrary domain. The limitations come both from failed parses and absurd interpretations that serve no useful purpose.

The current state of the art is that programs can perform very limited processing of fairly broad bodies of text as in news categorization, [Hayes *et al.*, 1988; Jacobs, 1992b], and somewhat more detailed processing of similar texts in constrained domains [Sundheim, 1991]. It is within reason to expect that work over the next several years will produce programs that can, for example, read a daily newspaper and extract useful, structured information from most items. The breadth and accuracy of parser coverage is clearly still in the critical path, as linguistic parsers tend to be of little help once the severe constraints of a domain are lifted.

As an example of the effects of these problems, the following is an analysis of a typical sentence selected from a corpus of *Wall Street Journal* text (November 2, 1989) obtained through the ACL Data Collection Initiative, along with some of the problems that *our* system had on its first pass at analyzing this corpus [Jacobs *et al.*, 1990]. We have informally verified that many other language analysis programs have most of the same problems.

The first sentence of the *Wall Street Journal* story is as follows:

> A form of asbestos once used to make Kent cigarette filters has caused a high percentage of cancer deaths among a group of workers exposed to it more than 30 years ago, researchers reported. (34 words)

This text is about typical in length and complexity for the first sentence of a news story, and was not from any domain that we had covered. While the system did produce some valid semantic information, such as that asbestos had caused the deaths, that cancer is a disease, and that the workers were somehow involved in dying, it encountered the following difficulties, which we consider to be representative:

- Failed parses and attachments, e.g., the phrase *to make Kent cigarette filters* was attached as a prepositional phrase modifying *form*, because *used* does not allow an infinitive complement.
- Extraneous considerations, e.g., the program found several parses with the noun phrase *Kent cigarette filters* broken up, with *used* as the main verb of the sentence or with *filters* as a verb.
- Inadequate depth of interpretation, e.g. the program correctly determined that *cancer* modified *deaths*, as well as the sense of *among*, but did not produce any representation of the fact that cancer somehow caused the deaths and that the workers died.

Clearly, some of these problems result from a lack of system knowledge; for example, the program did not recognize the word *asbestos* and should have been able to link diseases to deaths (since it did "know" that *cancer* is a disease). On the other hand, just as many of the problems result from *too much knowledge*—that is, the extra work and false interpretations stemming from the complexity of the system. These problems boil down to two paradigmatic issues—coverage and attachment.

The double-edged sword of coverage

Extending a parsing system's knowledge base avoids "gaps" and increases the range of inputs that can be interpreted, but also increases the combinatorics of parsing and the likelihood that the system will produce an incorrect interpretation. For example, knowing that *make* and *high* can be nouns seems necessary for handling real texts, but the system must discard such possibilities quickly from consideration in the *Kent cigarette* example. Similarly, determiners are optional in many prepositional phrases with the preposition *to*, but this knowledge can cause spurious parses of many infinitive phrases. Intuitively, disambiguating *to* at the sentence level doesn't seem to make sense, if *to lunch* is a thousand times more likely to be a prepositional phrase than *to make*. But this is precisely the sort of disambiguation most parsers try to do, often relying on sentence-level constraints where "weaker" lexical or domain knowledge would produce much better results.

The entrapment of attachment

Attachment, especially of prepositional phrases, has been the focus of much parsing work (e.g. [Frazier and Fodor, 1978; Schubert, 1986]) because it is the source of most of the combinatorics in parsing. Yet phrase attachment does not always contribute very much to the results of text analysis, particularly in data extraction. Even worse, an otherwise inconsequential attachment decision can actually interfere with getting the correct semantic interpretation.

Regardless of the degree to which syntax contributes to correct attachment, the combination of attachment possibilities is a misdirected parsing effort. In addition to phrases like *among a group of workers*, which can legitimately attach in three places, temporal modifiers like *more than 30 years ago* can attach almost anywhere. This sort of phrase is especially common in news stories. The result is that a typical parser spends more time deliberating where to attach phrases whose attachment *doesn't* help semantic interpretation than it does on phrases whose attachment *does* matter, and parsing is fundamentally misdirected.

The relation-driven control strategy, by loosely coupling linguistic processing and attachment to constituent structure, avoids problems with spurious lexical and at-

tachment decisions and allows these decisions to derive from a combination of knowledge sources.

3 Relation-driven control

At the heart of these issues is the problem that most full parsers are fundamentally clause-driven, from the extreme case of those that produce no information at all when they fail to recognize a sentence, to the more typical parsers that resolve attachment ambiguities at the sentence level. Many programs often can't tell if a word is a noun or a verb without building a complete sentence structure around it.

Relation-driven control is a model of parsing and semantic interpretation in which linguistic relations serve as the focal point of analysis. These relations, such as those between subject and verb, verb and complement, or noun and modifier, are *independent* of syntactic analysis in that they are not tied to a particular complex linguistic structure. While the relations are themselves linguistic structures, they associate easily with conceptual roles and statistical knowledge (such as co-ocurrence), making it easier to validate them without a full parse.

Figure 1 breaks down the major components of our system, showing how relation-driven control mediates among three general types of processing—(1) corpus-driven, shallow pre-processing, (2) linguistic analysis, including parsing and semantic interpretation, and (3) post-processing, mainly template-filling and other domain-driven processing. The term *template* here refers to the final result of data extraction—a frame with a set of conceptual features for each particular type of object or event (see Section 4). This model is implemented, although there is a great deal more work to be done, particularly in expanding the pre-processing and recovery components. The system uses the TRUMP [Jacobs, 1992c] analyzer, and is currently being extended to use Carnegie Mellon University's implementation of Tomita's algorithm [Tomita, 1986] for multi-lingual processing under the government-sponsored TIPSTER program.

A relation is a triple consisting of a head, a role, and a filler (or non-head). Each linguistic relation maps into a conceptual relation that associates word senses with conceptual roles, giving rise to three types of semantic preferences. Figure 2 illustrates a relation and the three types of preferences.

Relations can derive from lexical preferences and other patterns established during pre-processing, from the semantic interpretation of constituents during parsing, and from domain knowledge when recovering from failed parses. Relation-driven control focuses the interpretation task on determining a preference score for each sense of each content word in the text, along with the semantic and template roles associated with that word sense. During interpretation, the program drops linguistic structures with low relation scores. After syntactic processing, the matrix of preference scores produces a total that selects the preferred interpretation of the text, even where the system has failed to produce a complete parse.

The *role* preference shown in Figure 2 is a measure of how well the filler fills a role. This is where the equivalent of selectional restrictions, and most domain knowledge, comes into play. For example, foods make good fillers for the *patient* of eating activities. Such preferences can be expressed even at the level of individual senses; for example, *pilot* is a good filler for the *agent* role of flying activities.

The *rel* preference measures the salience of a role with respect to the head. For example, concepts such as monetary objects and units of measure "expect" to be qualified by some *quantity*. Thus, while the baseline preference for *yard* is for an enclosed area, the preferred interpretation of *ten yards* is as a measure. There is no reason why areas *shouldn't* be modified by quantities, but the role is much more salient for measures.

The *base* preference associates the filler with the head, independent of the particular role. For example, employment actions and professions tend to be related, regardless of roles. Thus *art employment* would typically mean the hiring of people in the art profession, rather than the use of works of art (or any of the other poor sense combinations). Distinguishing this sort of preference is necessary because the language very often provides little information about the conceptual role.

In the example shown in Figure 2, the sentence includes a free infinitive adjunct *to be acquired...*—a common sort of interpretation problem because the normal syntactic preferences would be to try to attach the phrase starting with the rightmost "node" (*agreement*). The role *target* is salient, and a company makes a good filler for the role, so both *role* and *rel* preferences are very high in the relation shown. The system thus favors any parse that will include this relation, and will attempt to retain the relation even if the parser fails.

The relation-driven control triangle supports fault tolerance, because strong preferences coming from any part of this data structure can determine the remaining portions. For example, a strong base preference often guides attachment decisions even in the absence of role preferences.

Most weights in the system are assigned automatically, either through statistical analysis in the case of collocations or by a knowledge-based rule that computes preference scores according to their position in the hierarchy. Thus a preference for *food* is automatically assigned a higher weight than a preference for *animate* because *food* is deeper in the hierarchy (i.e. the *weight* is assigned automatically; the *preference* itself is manually coded). This method is reasonably effective because the interpreter is most often comparing one or two combinations with preferences to a great many combinations with no preferences or negative preferences (violated restrictions).

The discussion that follows gives some more details and examples of the benefits of relation-driven control with respect to the corpus- and task-driven components

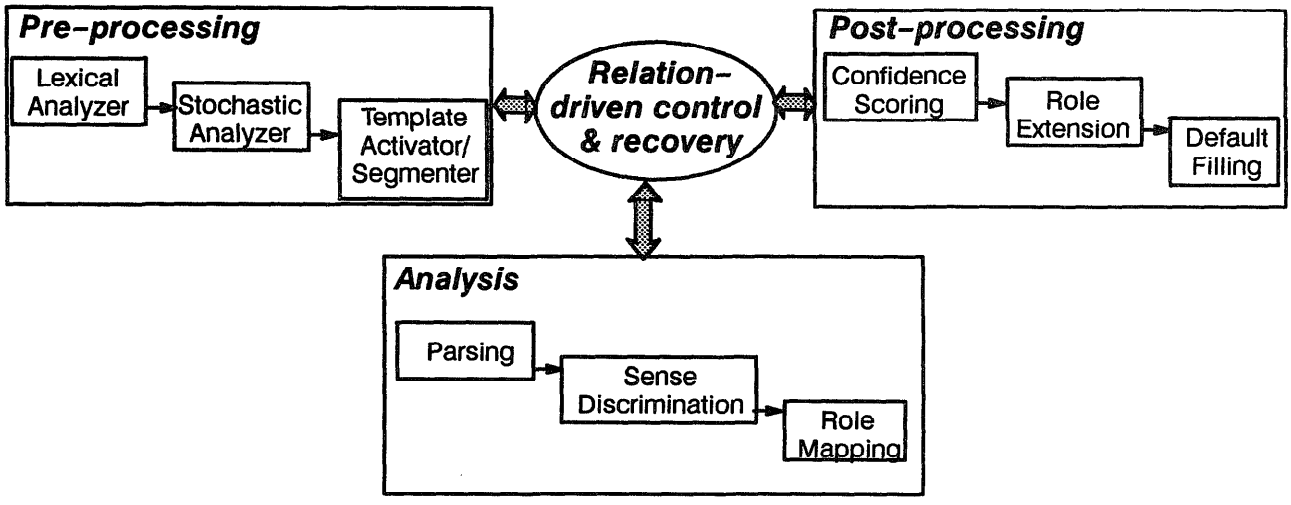

Figure 1: Stages of data extraction

of the system.

3.1 Pre-processing

Pre-processing helps to segment texts, control the interpretation of words in the text, and identify where phrase attachment is necessary. As Figure 1 shows, the relation-driven control strategy depends on key information that the system obtains prior to parsing, including lexical preferences and other "dynamic" lexical information, statistical results such as phrase acquisition [Jacobs, 1992a], and template activation and other domain-specific segmentation. [2]

The following is a sample text taken from the development corpus of the MUC-3 message understanding evaluation, with the results of the segmentation component of pre-processing:

Original text:

> SIX PEOPLE WERE KILLED AND FIVE WOUNDED TODAY IN A BOMB ATTACK THAT DESTROYED A PEASANT HOME IN THE TOWN OF QUINCHIA, ABOUT 300 KM WEST OF BOGOTA, IN THE COFFEE-GROWING DEPARTMENT OF RISARALDA, QUINCHIA MAYOR SAUL BOTERO HAS REPORTED. (41 words)

Segmented text:

> [SIX PEOPLE] [WERE KILLED] AND FIVE [WOUNDED] [TIME: TODAY] [IN A BOMB ATTACK] THAT [DESTROYED] [A PEASANT HOME] [LOCATION: IN THE TOWN OF QUINCHIA] [DISTANCE: *COMMA* ABOUT 300 KM WEST OF BOGOTA] [LOCATION: *COMMA* IN THE COFFEE *HYPHEN* GROWING DEPARTMENT OF RISARALDA] [SOURCE: *COMMA* QUINCHIA MAYOR SAUL BOTERO HAS REPORTED] *PERIOD*

The task in processing these examples is to fill, for each news story, a set of possible templates, each containing 18 fields indicating the type of event, perpetrator(s), victim(s), etc.

By grouping and labeling portions of text early, the program greatly reduces the amount of real parsing that must be done, eliminates many failed parses, and provides template-filling information that helps with later processing. For example, the phrase *in the town of Quinchia* is at least five ways ambiguous—it could modify *a peasant home*, *destroyed*, *a bomb attack*, *wounded*, or *were killed and five [were] wounded*. However, all five of these possibilities have the same effect on the final templates produced, so the program can defer any decisions about how to parse these phrases until after it has determined that the killing, wounding, attacking, and destruction are all part of the same event. Since these choices combine with the ambiguity of other phrases, the parsing process would otherwise be needlessly combinatoric. In fact, parsing contributes nothing after *a peasant home*, so this sentence can be processed as a 16-word example with some extra modifiers.

In addition to reducing the combinatorics of modifier attachment, this pre-processing helps in resolving false ambiguities that are a matter of style in this sort of text. In this example, the ellipsis in *five [were] wounded* would be difficult, except that *wounded*, like many transitive verbs, is never used as an active verb without a di-

[2] The identification of segments of texts that activate and fill templates is described in [Jacobs, 1990]

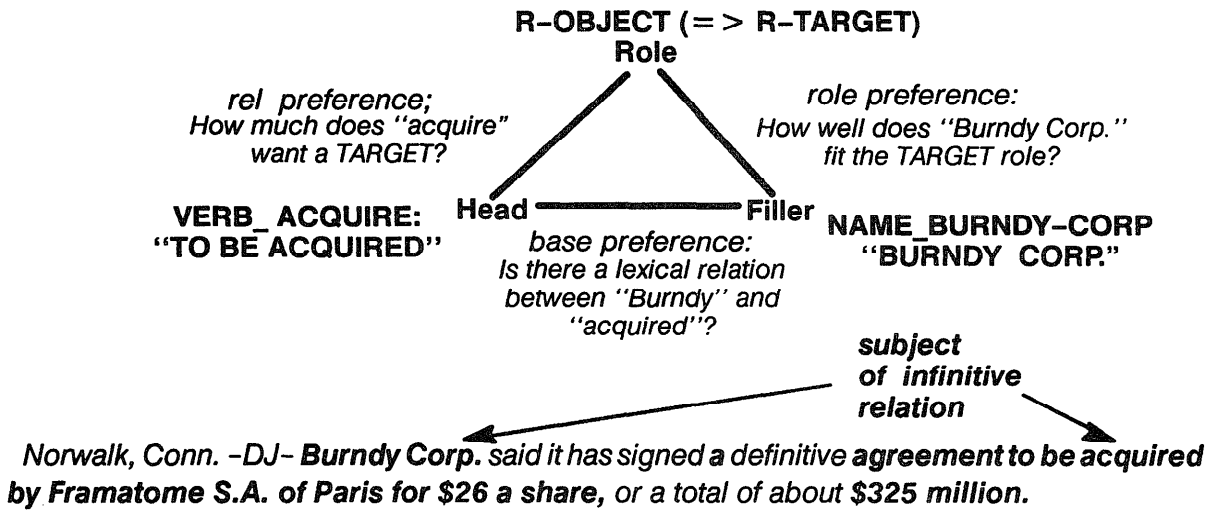

Figure 2: Relations and semantic preferences

rect object. The early bracketing of the text allows the parser, through the relation-driven control module, to resolve the common past participle–past tense ambiguities without having to wait for a complete verb phrase.

3.2 Domain-driven processing

The integration of domain-driven analysis and linguistic analysis is probably the hardest and most critical problem for data extraction systems, as explained in [Rau and Jacobs, 1988] and [Passonneau *et al.*, 1990]. With relation-driven control, the system looks for linguistic relations in the text that affect template filling whether or not it succeeds in producing a complete parse, and avoids relations that do not influence template filling. The segmentation of the text during pre-processing guides this process. Relation-driven control uses the template activators, typically verbs, as "pivots", and looks for relations (such as subject-verb and verb-object) that might include each pivot. These attachments do not wait for a complete parse, so this method is more robust, and favors attachments that are valid in the domain.

Relation-driven control depends on having a rough structure of the events of the text prior to parsing. This is realistic in most domains we have dealt with, including the terrorism stories, where death and destruction are subsidiary to attacks and bombings. [3] The event structure determines what roles must be filled linguistically, avoiding attachment of shared roles (such as instrument, time and location) and favoring the attachment of valid roles.

In this example, the program produces the templates in Figure 3.

[3] In corporate mergers and acquisitions [Jacobs and Rau, 1990a], rumors, offers, and stock moves are subsidiary takeover events.

This section has addressed the principal issues in focusing parsing to resolve the coverage and attachment problems described in Section 2. The next section will present some preliminary results from this broad-scale effort.

4 System evaluation and status

In earlier projects, such as SCISOR [Jacobs and Rau, 1990a] and other efforts involving the GE NLToolset [Jacobs and Rau, 1990b], our programs produced accuracy of over 80% in data extraction from texts in limited domains (for example, getting an average of 80% of the suitors correct in stories about mergers and acquisitions) and have compared very favorably to other systems in evaluations on a common task. The system was in the top group in every measure in the MUC-3 evaluation [Krupka *et al.*, 1991a; Lehnert and Sundheim, 1991] (which uses a much harsher scoring method), while producing half as many templates as most of the other top systems. The program increased in processing speed over a two-year interval by a factor of 12, to over 1000 words/minute. Most importantly, the techniques described here helped achieve accuracy in MUC-3 comparable to the MUCK-II evaluation two years earlier, although an analysis showed MUC-3 at least an order of magnitude broader [Hirschman, 1991] along several dimensions.

Having been tested in an early version in MUC-3, the relation-driven control strategy is a major component of the GE-CMU team's technical strategy in the DARPA TIPSTER–Data Extraction project. Interestingly, this strategy is the closest to traditional parsing of all the TIPSTER teams, with other groups relying more heavily on methods such as probabilistic parsing and phrase-based domain-driven analysis. One of the major moti-

```
0.  MESSAGE ID                          DEV-MUC3-0644
1.  TEMPLATE ID                         1
2.  DATE OF INCIDENT                    07 NOV 89
3.  TYPE OF INCIDENT                    BOMBING
4.  CATEGORY OF INCIDENT                TERRORISM
5.  PERPETRATOR: ID OF INDIV(S)         -
6.  PERPETRATOR: ID OF ORG(S)           -
7.  PERPETRATOR: CONFIDENCE             -
8.  PHYSICAL TARGET: ID(S)              "PEASANT HOME"
9.  PHYSICAL TARGET: TOTAL NUM          1
10. PHYSICAL TARGET: TYPE(S)            OTHER
11. HUMAN TARGET: ID(S)                 "PEOPLE"
12. HUMAN TARGET: TOTAL NUM             11
13. HUMAN TARGET: TYPE(S)               CIVILIAN
14. TARGET: FOREIGN NATION(S)           -
15. INSTRUMENT: TYPE(S)                 *
16. LOCATION OF INCIDENT                COLOMBIA: QUINCHIA (CITY):
                                                RISARALDA (DEPARTMENT)
17. EFFECT ON PHYSICAL TARGET(S)        SOME DAMAGE
18. EFFECT ON HUMAN TARGET(S)           DEATH
                                        INJURY

0.  MESSAGE ID                          DEV-MUC3-0644 (GE)
1.  TEMPLATE ID                         2
2.  DATE OF INCIDENT                    07 NOV 89
3.  TYPE OF INCIDENT                    MURDER
4.  CATEGORY OF INCIDENT                TERRORISM
5.  PERPETRATOR: ID OF INDIV(S)         -
6.  PERPETRATOR: ID OF ORG(S)           -
7.  PERPETRATOR: CONFIDENCE             -
8.  PHYSICAL TARGET: ID(S)              *
9.  PHYSICAL TARGET: TOTAL NUM          *
10. PHYSICAL TARGET: TYPE(S)            *
11. HUMAN TARGET: ID(S)                 "PEOPLE"
12. HUMAN TARGET: TOTAL NUM             6
13. HUMAN TARGET: TYPE(S)               CIVILIAN
14. TARGET: FOREIGN NATION(S)           -
15. INSTRUMENT: TYPE(S)                 BOMB
16. LOCATION OF INCIDENT                COLOMBIA: QUINCHIA (CITY):
                                                RISARALDA (DEPARTMENT)
17. EFFECT ON PHYSICAL TARGET(S)        *
18. EFFECT ON HUMAN TARGET(S)           *
```

Figure 3: Results of Data Extraction

vations for the emphasis on control strategy in the GE-CMU team has been to take advantage of the established framework of the CMU Generalized LR Parser within the context of GE's text processing system. This project is barely underway; however, the system has already processed its first message set after the integration of the CMU parser with a GE grammar and pre-processor. The combination of systems of this scope in linguistic analysis is, to our knowledge, unprecedented, and illustrates the growing importance of control strategies for combining different analysis technologies.

5 Conclusion

The relation-driven control strategy directs linguistic processing toward identifying and disambiguating relations centering on relevant portions of text, instead of on completing and attaching syntactic constituents. This strategy eliminates much of the needless combinatorics of parsing without ignoring the syntactic constraints and preferences that affect semantic results. This ongoing work has produced an efficient parser with broad coverage that has been applied to large quantities of real text, with some promising results in word sense discrimination and data extraction.

References

[Church *et al.*, 1989] K. Church, W. Gale, P. Hanks, and D. Hindle. Parsing, word associations, and predicate-argument relations. In *Proceedings of the International Workshop on Parsing Technologies*, Carnegie Mellon University, 1989.

[Frazier and Fodor, 1978] L. Frazier and J. D. Fodor. The sausage machine: A new two-stage parsing model. *Cognition*, 6:291–325, 1978.

[Hayes *et al.*, 1988] Phil Hayes, Laura Knecht, and Monica Cellio. A news story categorization system. In *Proceedings of Second Conference on Applied Natural Language Processing*, Austin, Tx, February 1988. Association for Computational Linguistics.

[Hirschman, 1991] L. Hirschman. Comparing muck-ii and muc-3: Notes on evaluating the difficulty of different tasks. In *Proceedings of the Third Message Understanding Conference (MUC-3)*, San Mateo, CA, May 1991. Morgan Kaufmann Publishers.

[Jacobs and Rau, 1990a] Paul Jacobs and Lisa Rau. SCISOR: Extracting information from on-line news. *Communications of the Association for Computing Machinery*, 33(11):88–97, November 1990.

[Jacobs and Rau, 1990b] Paul S. Jacobs and Lisa F. Rau. The GE NLToolset: A software foundation for intelligent text processing. In *Proceedings of the Thirteenth International Conference on Computational Linguistics*, volume 3, pages 373–377, Helsinki, Finland, 1990.

[Jacobs *et al.*, 1990] Paul S. Jacobs, George R. Krupka, Susan W. McRoy, Lisa F. Rau, Norman K. Sondheimer, and Uri Zernik. Generic text processing: A progress report. In *Proceedings of the Third DARPA Speech and Natural Language Workshop*, Somerset, PA, June 1990. Morgan Kaufmann.

[Jacobs, 1990] Paul Jacobs. To parse or not to parse: Relation-driven text skimming. In *Proceedings of the Thirteenth International Conference on Computational Linguistics*, pages 194–198, Helsinki, Finland, 1990.

[Jacobs, 1992a] Paul S. Jacobs. Acquiring phrases statistically. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*, Newark, Delaware, 1992. Submitted.

[Jacobs, 1992b] Paul S. Jacobs. Joining statistics with NLP for text categorization. In *Proceedings of the 3rd Conference on Applied Natural Language Processing*, April 1992.

[Jacobs, 1992c] Paul S. Jacobs. TRUMP: A transportable language understanding program. *International Journal of Intelligent Systems*, 7(3):245–276, 1992.

[Krupka *et al.*, 1991a] George R. Krupka, Łucja Iwańska, Paul S. Jacobs, and Lisa F. Rau. GE NLToolset: Muc-3 test results and analysis. In *Proceedings of the Third Message Understanding Conference (Muc-3)*, San Mateo, Ca, May 1991. Morgan Kaufmann Publishers.

[Krupka *et al.*, 1991b] George R. Krupka, Paul S. Jacobs, Lisa F. Rau, and Łucja Iwańska. Description of the GE NLToolset system as used for Muc-3. In *Proceedings of the Third Message Understanding Conference (Muc-3)*, San Mateo, Ca, May 1991. Morgan Kaufmann Publishers.

[Lehnert and Sundheim, 1991] Wendy G. Lehnert and Beth M. Sundheim. A performance evaluation of text analysis technologies. *Artificial Intelligence Magazine*, 12(3):81–94, Fall 1991.

[Passonneau *et al.*, 1990] R. Passonneau, C. Weir, T. Finin, and M. Palmer. Integrating natural language processing and knowledge based processing. In *Proceedings of the Eighth National Conference on Artificial Intelligence*, pages 976–983, Boston, 1990.

[Rau and Jacobs, 1988] Lisa F. Rau and Paul S. Jacobs. Integrating top-down and bottom-up strategies in a text processing system. In *Proceedings of Second Conference on Applied Natural Language Processing*, pages 129–135, Austin, TX, Feb 1988. ACL.

[Schubert, 1986] L. Schubert. Are there preference trade-offs in attachment decisions? In *Proceedings of the Fifth National Conference On Artificial Intelligence*, 1986.

[Sharman *et al.*, 1990] R. Sharman, F. Jellinek, and R. Mercer. Generating a grammar for statistical training. In *Darpa Speech and Natural Language Workshop*, Hidden Valley, Pa, 1990.

[Sundheim, 1989] Beth Sundheim. Second message understanding (MUCK-II) report. Technical Report 1328, Naval Ocean Systems Center, San Diego, Ca, 1989.

[Sundheim, 1991] Beth Sundheim, editor. *Proceedings of the Third Message Understanding Conference (Muc-3)*. Morgan Kaufmann Publishers, San Mateo, Ca, May 1991.

[Tomita, 1986] M. Tomita. *Efficient Parsing for Natural Language*. Kluwer Academic Publishers, Hingham, Massachusetts, 1986.