The GE NLToolset: A Software Foundation for Intelligent Text Processing

Paul S. Jacobs and Lisa F. Rau
Artificial Intelligence Program
GE Research and Development Center
Schenectady, NY 12301 USA
rau@crd.ge.com, psjacobs@crd.ge.com

Many obstacles stand in the way of computer programs that could read and digest volumes of natural language text. The foremost of these difficulties is the quantity and variety of knowledge about language and about the world that seems to be a prerequisite for any substantial language understanding. In its most general form, the robust text processing problem remains insurmountable; yet practical applications of text processing are realizable through a combination of knowledge representation and language analysis strategies.

This project note describes the GE NLTOOLSET and its use in two text processing applications. In the first domain, the system selects and analyzes stories about corporate mergers and acquisitions as they come across a real-time news feed. In the second domain, the program uses naval operations messages to fill a 10-field template. In both cases, users can ask natural language questions about the contents of the texts, and the system responds with direct answers along with the original text.

The GE NLTOOLSET is a software foundation for text processing. The NLTOOLSET derives from a research effort aimed at preserving the capabilities of natural language text processing across domains. The program achieves this transportability by using a core knowledge base and lexicon that customizes easily to new applications, along with a flexible text processing strategy tolerant of gaps in the program's knowledge base. Developed over the last four years, it runs in real time on a SUN^{TM} workstation in Common Lisp under UNIX^{TM}. It performs the following tasks:

- The lexical analysis of the input character stream, including names, dates, numbers, and contractions.
- The separation of the raw news feed into story structures, with separate headline, byline and dateline designations.
- A topic determination for each story, indicating whether it is about a corporate merger.
- The natural language analysis of each selected story using an integration of two interpretation strategies—"bottom-up" linguistic analysis and "top-down" conceptual interpretation.
- The storage and retrieval of conceptual representations of the processed texts into and out of a knowledge base.

The design of the NLTOOLSET combines artificial intelligence (AI) methods, especially natural language processing, knowledge representation, and information retrieval techniques, with more robust but superficial methods, such as lexical analysis and word-based text search. This approach provides the broad functionality of AI systems without sacrificing robustness or processing speed. In fact, the system has a throughput for real text greater than any other text extraction system we have seen (e.g., [Sondheimer, 1986; Sundheim, 1990]), while providing knowledge-based capabilities such as producing answers to English questions and identifying key conceptual roles in the text (such as the suitor, target, and per-share price of a merger offer). The NLTOOLSET consists of roughly 50,000 lines of Common Lisp code. It was developed entirely on SUN workstations.

1 Technical Overview

The NLTOOLSET's design provides each system component with access to a rich hand-coded knowledge base, but each component applies the knowledge selectively, avoiding the computation that a complete analysis of each text would require. The architecture of the system allows for *levels* of language analysis, from rough skimming [Jacobs, 1990] to in-depth conceptual interpretation [Jacobs, 1987].

A custom-built 10,000 word-root lexicon and concept hierarchy provides a rich source of lexical information. Entries are separated by their senses, and contain special context clues to help in the sense-disambiguation process. A morphological analyzer contains semantics for about 75 affixes, and can automatically derive the meanings of inflected entries not separately represented in the lexicon. Domain-specific words and phrases are added to the lexicon by connecting them to higher-level concepts and categories present in the system's core lexicon and concept hierarchy. This is one aspect of the NLTOOLSET that makes it highly portable from one domain to another.

The language analysis strategy used in the NLTOOLSET combines full syntactic (bottom-up) parsing and conceptual expectation-driven (top-down) parsing. Four knowledge sources, including syntactic and semantic information and domain knowledge, interact in a flexible manner. This integration produces a more robust semantic analyzer that deals gracefully with gaps in lexical and syntactic knowledge, trans-

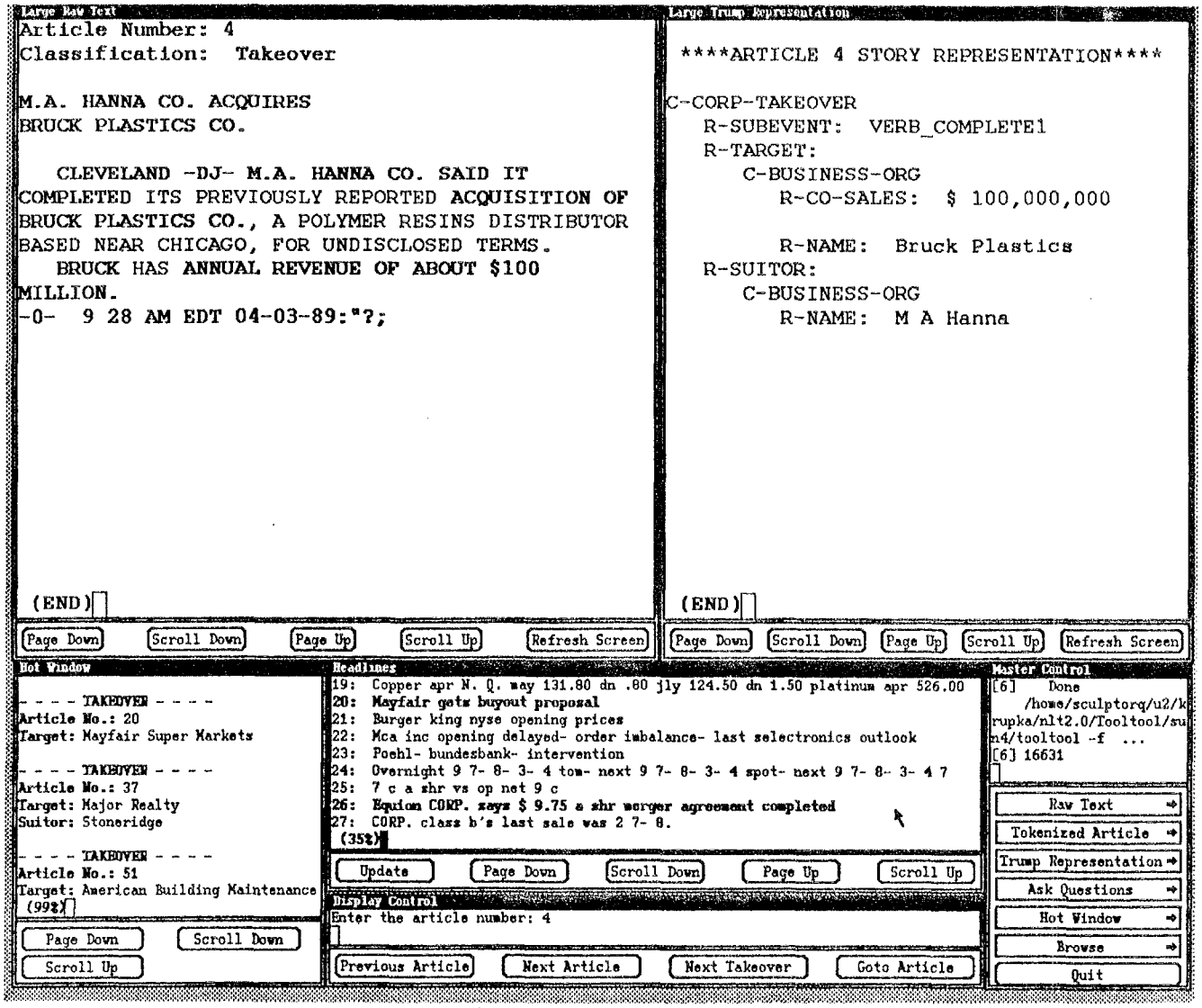

Figure 1: SCISOR in action

ports easily to new domains, and facilitates the extraction of information from texts [Rau and Jacobs, 1988].

Two prototype systems (both to be demonstrated at Coling) illustrate some of the capabilities of the NLTOOLSET. SCISOR (System for Conceptual Information Summarization, Organization, and Retrieval) reads financial news stories from a news service, selects stories about mergers and acquisitions, extracts key pieces of information from those stories, and answers English questions about this information. MUCK-II (a demonstration from a message understanding conference in 1989 [Sundheim, 1990]) shows some of the same capabilities, including database generation, question answering, and automatic alert, applied to a set of naval messages (OPREP-3). Both systems process texts at a rate of hundreds of paragraphs per hour. The customization of the NLTOOLSET to the MUCK-II application, porting from the domain of corporate takeovers to naval operations, required only several weeks.

2 SCISOR

SCISOR is a customization of the NLTOOLSET to the domain of news stories about mergers and acquisitions. The program analyzes stories as they come across a live news feed, selecting the takeover stories and applying a combination of top-down and bottom-up language analysis to identify conceptual roles in the stories. The result of this analysis is a single representation of each story that the program adds to a central knowledge base. The conceptual retrieval component accesses information from this knowledge base by analyzing English questions in the same manner and matching the questions to the story representations stored in the knowledge base.

SCISOR provides the user with information in multiple forms. Users can browse the headlines and the original texts. A "hot window" continuously displays the target, suitor, and price of the latest takeover stories, and flashes when a new takeover story comes across the wire. For more general information needs, an "ask question" window allows the user to type in simple English questions (e. g., "What was offered for Polaroid?") as well as query fragments (e. g.,

"acquisitions by Shamrock").

Figure 1 shows a SUN screen during the operation of SCISOR. The "Master Control" window in the lower right allows the user to open or access the various features of the system. The "Headlines" and "Display Control" in the lower center show the headlines of all stories (with headlines of takeover stories in bold) and guide the selection of texts for browsing. The "Hot Window", or alert feature, is at the lower left, alerting users the instant a new, potentially relevant article comes across the news wire. The "Raw Text" and "Trump Representation" windows at the top display each selected story, showing key portions of text in boldface with a summary of the language analysis in the upper right.

More details on the system design and operation of SCISOR can be found in [Jacobs and Rau, 1990].

3 Performance Evalution

Performance evaluation of natural language systems is a new problem, although the evaluation methods can adopt some of the techniques of traditional information retrieval (IR) systems. It would be difficult and probably futile to perform a controlled study of the NLTOOLSET against a traditional IR system, for two reasons: (1) traditional IR systems are tested on arbitrary, unconstrained texts, while natural language systems still work only in constrained domains; (2) the NLTOOLSET performs many tasks other than document retrieval, such as extracting information from stories and directly answering users' questions.

Evaluation problems of the entire system stem from the unique functionality of the NLTOOLSET system. Document retrieval systems, even sophisticated ones like RUBRIC[Tong *et al.*, 1986], do not extract features from from the documents they retrieve; thus it is impossible to compare them to NLTOOLSET. However, we have performed some tests that do measure the NLTOOLSET's accuracy in specific tasks.

The government-sponsored MUCK-II evaluation is, to our knowledge, the most meaningful test of natural language text processing, but the participants in the MUCK-II evaluation agreed not to release the specific results of the experiment. However, we will try to summarize the status of performance evaluation in general terms. Evaluation of content-based text processing systems like SCISOR is not nearly as established as evaluation methods in information retrieval. There are many tasks to be tested in this emerging type of system, including accuracy of question answering, helpfulness of alerts, and coverage of structured information (such as target and suitor). No mature methods exist for testing any of these tasks.

In spite of the problems with evaluating this sort of system, we would like to be informative about how our program performs. As a rule, it can extract key features from large sets of constrained texts with 80-90% (combined recall and precision) accuracy. It can achieve better results (and has) with more constrained texts, but would also produce almost nothing useful, say, in reading the entire *Wall Street Journal*. It is realistic to expect 90% accuracy for certain useful, carefully-constructed tasks, and unrealistic to expect much higher than this[1]. Many difficulties in reading texts appear when trying to achieve better results, but the most common limitation seems to be the degree of real inference required for understanding. In spite of its fairly sophisticated methods for combining linguistic and world knowledge, the NLTOOLSET really has very little of the latter.

In a recent test of SCISOR, the program analyzed one day's worth of stories directly from the newswire source. Of the 729 stories, the filter achieved slightly over 90% averaged recall and precision in its determination of which stories were about mergers and acquisitions (69 in all). SCISOR correctly identified the target and suitor in 90% of all the stories. When dollar-per-share amounts of offers were present in the stories, SCISOR extracted this quantity correctly 79% of the time, and the total value of the offer 82% of the time.

References

[DeJong, 1979] Gerald DeJong. Prediction and substantiation: A new approach to natural language processing. *Cognitive Science*, 3(3):251–273, 1979.

[Jacobs and Rau, 1990] Paul Jacobs and Lisa Rau. SCISOR: A system for extracting information from on-line news. *Communications of the Association for Computing Machinery*, 35, *(In Submission)* 1990.

[Jacobs, 1987] Paul S. Jacobs. A knowledge framework for natural language analysis. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, Milan, Italy, 1987.

[Jacobs, 1990] P. Jacobs. To parse or not to parse: Relation-driven text skimming. In *Proceedings of the Thirteenth International Conference on Computational Linguistics*, Helsinki, Finland, 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, Morristown, NJ, Feb 1988. ACL.

[Sondheimer, 1986] N Sondheimer. Proceedings of DARPA's 1986 strategic computing natural language processing workshop. Technical Report ISI/SR-86-172, University of Southern California, ISI, 1986.

[Sundheim, 1990] Beth Sundheim. Second message understanding conference (MUCK-II) test report. Technical Report 1328, Naval Ocean Systems Center, San Diego, CA, 1990.

[Tong *et al.*, 1986] Richard M. Tong, L. A. Appelbaum, V. N. Askman, and J. F. Cunningham. RUBRIC III: An object-oriented expert system for information retrieval. In *Proceedings of the 2nd Annual IEEE Symposium on Expert Systems in Government*, Washington, DC., October 1986. IEEE Computer Society Press.

[1] The FRUMP[DeJong, 1979] program, for comparison purposes, achieved 38% accuracy in one test on newswire stories.