

The FINITE STRING Newsletter

Site Report

Computational Linguistics Research at Duke University

Faculty: Bruce Ballard, Alan Biermann

Consultants: Robert Rodman (Ph.D. in Linguistics), David Rubin (Ph.D. in Psychology), Fran Heidlage (Ph.D. in Physiology)

Students: William Cohen, Amr Fahmy, Linda Fineman, Pamela Fink, Casey Gilbert, Gary Jackoway, Barry Koster, John Lusth, Karen Stone, Nancy Tinkham

1. Introduction

Research in natural language (NL) processing at Duke University began in the summer of 1977 with the design of the Natural Language Computer (NLC, an extant automatic programming system that provides an English-language *programming* environment for matrix problems. An initial implementation of NLC was completed early in 1979, and a report on this first version of the system appeared in *AJCL* the following year (Biermann and Ballard 1980).

Since 1979 the number of persons doing computational linguistics research at Duke has grown from three to somewhere over a dozen. During this time our goals have grown to include interests in

- (a) the design of *task-oriented* NL processors;
- (b) facilities to respond to *voice inputs*;
- (c) methods that enable an NL system to be easily *customized* for new applications or for new users;
- (d) *human factors testing* of NL processors; and
- (e) the development of various *formalisms* that will prove useful outside the context of the Duke systems.

We summarize our activities in each of these areas below.

2. Task-Oriented Systems

This work assumes a user is seated behind a computer display with a problem to be solved. The objects related to the task domain are visible on the screen and can be manipulated via natural language commands. The effect of each user input is immediately indicated on the screen, so the user can continuously verify that correct action is taking place. If an undesired result is noticed, the user can request a backup and then rephrase the command.

The first task-oriented natural language system built at Duke, the NLC, was designed to do matrix calculations. A user of NLC can request the display of one or several matrices, enter data into them, label rows and

columns, and perform a variety of arithmetic operations. For example, if a certain matrix is currently in focus, the user might say

“Add the sum of the positive entries in row 1 to the last negative element of the matrix.”

In addition to providing a fairly broad coverage of pronouns and conjunctions, NLC includes special “programming” features to allow for procedure definition (Ballard 1979), loops (Sigmon 1981), and conditionals (Fink 1983).

Our second task-oriented system, VIPS, represents a greatly simplified redesign of NLC and is aimed at office automation applications (Biermann and Ballard 1983). VIPS has been implemented to handle text manipulations from voice commands supplemented by touch inputs. For example,

“Put this sentence [touch] after the last sentence in this [touch] paragraph.”

Touch inputs to VIPS are currently handled via a Carroll 19-inch high-resolution color display unit with touch-sensitive screen.

Current work related to task-oriented systems includes the evolution of a “micro-model” of a generic processor which will enable precise statements to be made concerning the construction and capabilities of specific systems.

3. Voice-Interactive Processing

Based upon our experience with typed English inputs, we became interested in developing high quality real-time voice interactive natural language processors. In 1982, we purchased a Nippon DP-200 connected speech recognizer and a Votan V-5000 speech recognizer and voice response unit, and have invested considerable effort in learning their characteristics. During this period, the VIPS system mentioned above was designed, and NLC was redesigned for voice input. A tape of the resulting VNLC system was played last year at the Applied Natural Language meeting in Santa Monica (Biermann et al. 1983) and the VNLC system was given its first public demonstration at an ACM regional meeting in April 1983.

More recently we have begun to work with a Verbox 3000 continuous speech recognizer, and some interesting research questions have arisen. For instance, we are investigating the possible advantages of “discrete” speech, where a 300 millisecond pause must follow every word, over the more natural “connected” speech, where word boundaries may merge. Though slow and unnatural, discrete speech has the advantage

that its increased recognizability makes it possible to use larger vocabularies and still maintain robust behavior. Connected speech allows much faster and more comfortable input, but with an accompanying loss of reliability and/or vocabulary size.

4. Adaptability and Transportability

Although several natural language processors provide *customization* facilities that allow users to specify synonyms, syntactic paraphrases, and the like, most existing systems are restricted to a particular *domain* of data to be accessed. Thus, users are unable to access novel data without acquiring a new or modified processor specifically tailored to the new domain from the system designer(s). For this reason, we have become interested in allowing users to adapt an existing system for a new domain of data. In addition to overt customization by users, we have also been interested in how a system might adapt itself to the style and needs of an individual user.

4.1 The layered domain class (LDC) system

The Layered Domain Class system (LDC) seeks to allow users of a natural language system to customize a processor to work with types of data unknown to the initial system designers (Ballard, Lusth and Tinkham 1984a, 1984b). A secondary but also important feature of LDC is its ability to work with loosely-structured input files, as opposed to more formal structures required by conventional database management systems and assumed by typical NL front-ends. Besides the obvious technological advantage of a user-customizable system, in terms of time and cost needed to acquire a specialized processor, efforts at providing transportable systems have scientific value in that they disallow ad hoc solutions to seemingly domain-specific problems in favor of methods based upon an understanding of the relation between the conceptual structure of a domain and the mechanisms needed to process the language used in discussing it.

To render the customization problem tractable, LDC restricts itself to "layered" domains (Ballard 1982), which emphasize *containment* relationships among domain objects and thus generalize upon the domains of LDC and VIPS. The initial interaction between a user and LDC, which involves telling the system about a new domain, consists of a knowledge-acquisition session with the preprocessor, which we call "Prep". In particular, Prep asks for

- (1) the names of each type of "entity" (object) of the domain;
- (2) the nature of the relationships among entities;
- (3) the English words that will be used as nouns, verbs, and modifiers; and
- (4) morphological and semantic properties of these new words.

Having completed a session with the user, Prep digests its newly acquired information to produce files that will be used during subsequent processing of English inputs.

Among the primary tasks to be undertaken in the foreseeable future are

- (a) implementation of certain exotic and anaphoric modifier types, as described in Ballard 1984;
- (b) user testing of the existing system, both in customization and in user mode;
- (c) provisions for limited use of coordinate conjunction;
- (d) introduction of a generic hierarchy (taxonomy) so that, for example, a "large office" can be assumed to be a large room that is also an office, unless special provisions have been made to the effect that "large" applies differently to offices than to other types of rooms; and
- (e) incorporation of the voice recognition capabilities developed for VNLC and VIPS.

4.2 Automatic adaptation to new users

Dialogues of the type that occur when one calls a travel agent, gives a weather report, or instructs a secretary in a routine task often manifest a repetitive or stereotypic structure, especially when the utterances of a single speaker are being considered. We call the set of all dialogues that may occur in a given environment a "dialogue type". While individual dialogues within a dialogue type may differ in content and style, they often have rather noticeable resemblances. For instance, reasonably stereotyped utterances may open a dialogue, convey significant facts, move from one subject to another, and close the interaction. The exact wording and ordering of sentences may change from one dialogue to another, but often enough a basic pattern can be identified and used to create an "expectation" as to which of several possible continuations (e.g. at some point in a sentence or dialogue) are most likely for the current user.

With these thoughts in mind, we have attempted to utilize such information to enable error correction in the context of on-line voice-interactive natural language processors of the type discussed above. In particular, we have devised methods that monitor incoming inputs to a voice-interactive natural language system and build up a graphical structure to capture the relevant portions of a dialogue type. This frame-like structure is then used as a predictor when a suspected voice recognition error has occurred. Experience indicates that speakers can in fact speak more rapidly (and less carefully) and still obtain acceptable recognition rates. Thoughts are being given to more liberal ways of utilizing the expectations built up during sessions with a user.

5. Human Factors Testing

The system development work described above has been accompanied by a series of tests to determine how well users can actually use a prototype system to solve real problems. For instance, we have investigated the degree of training required to make productive use of task-oriented systems, how *habitable* the presumably broad vocabulary and grammar turns out to be, how *robust* the processor is, and the *reaction* of users to the experience of using an English language system. For the most part, our results have been encouraging. For example, in one user study of NLC (Biermann, Ballard and Sigmon 1983), it was found that subjects had learned to use the system quite effectively after only 50 minutes of training. The system responded correctly to about 81 percent of all user inputs, and people seemed to enjoy working with the system. Another interesting performance study gives information about the usefulness of NLC by students in a sophomore-level linear algebra course at Duke (Geist, Kraines, and Fink 1983).

One important observation that came out of our human factors work was that subjects spent an unexpectedly large portion of their time typing the inputs. Thus a typical input-output cycle lasted about 50 seconds: 20 seconds for typing, 3 second waiting for a computer response, and the remaining 27 seconds thinking about what to do next. On the basis of this, we decided to purchase the voice equipment mentioned above. The effect has been both to shorten the time required to input a command and to encourage continuous eye contact with the objects being manipulated. In fact, users now input sentences at least twice as fast with voice input than with typed input.

6. Formalisms for NL System Design

In the course of designing the processors summarized above, we have developed a number of *formalisms*, i.e. specification languages, intended for use outside the immediate environment of the Duke systems. Our primary intention has been to reduce and simplify the effort needed to mediate between the syntactic and retrieval portions of a natural language interface. Toward this goal, we have developed a phrase-structured grammatical formalism and a high-powered but domain-independent retrieval query language.

6.1 A phrase-structured grammatical formalism

Seeking to take advantage of the benefits of both the familiar ATN formalism of Woods and the augmented phrase-structured grammars of Heidorn, we have developed a hybrid grammatical formalism based on *augmented phrase-structure* rules that allows a simple but general parser to make its domain-specific decisions by reference to auxiliary files produced during a learning session of the sort carried out during the knowledge

acquisition phase of LDC. Our grammatical formalism (Ballard and Tinkham 1984) is built around seven command types. Three of these are used to specify words, parts of speech, and syntactic categories, while the remaining four provide control facilities for optionality, possible repetition, alternation, and sequence. In addition to the main function of each of the commands, through which the grammar writer can specify any context-free grammar, most commands allow for various forms of augmentations, useful in specifying the constraints needed for the parser to perform useful disambiguations. We are currently engaged in a redesign of our parser to be run on a Symbolics 3670.

6.2 A high-level retrieval query language

In the context of our work on the LDC system described above, we have abandoned the common practice of building a front end to an pre-existing retrieval system (e.g. DBMS). Instead, we have sought to develop methods whereby the *sophisticated* processing required by English inputs can be handled *directly* by a retrieval component. Roughly speaking, this allows an English processing component to concentrate upon syntactic matters, and relegate matters of semantic processing to the database retrieval component, where ideally they belong.

To accomplish this, we have designed a powerful but simple query language called DOMINO (Lusth and Ballard 1984), some of whose novel features are

- (1) facilities to access text-edited files, as opposed to more restrictive database structures,
- (2) several high-powered operators to loop through selected portions of the data file, with arbitrary levels of nesting permitted, and
- (3) capabilities for "macro" specifications by users, wherein arbitrarily complex operations can be entered once, either before or during a session, and subsequently accessed in a single step, at any meaningful place within a query, just as though they had been supplied by the initial retrieval processor.

Current work with DOMINO involves the addition of further and still more powerful built-in looping capabilities, anaphoric macro calls, and a possible redesign for the Symbolics environment.

References

- Ballard, B. 1979 Semantic and Procedural Processing for a Natural Language Programming System (Ph.D. dissertation). Technical Report CS-1979-5, Department of Computer Science, Duke University, Durham, North Carolina.
- Ballard, B. 1982 A Domain-Class Approach to Transportable Natural Language Processing. *Cognition and Brain Theory* 5 (3): 269-287.
- Ballard, B. 1984 The Syntax and Semantics of User-Defined Modifiers in a Transportable Natural Language Processor. *Proceedings of COLING 84*.

Ballard, B. and Lusth, J. 1983 An English-Language Processing System That "Learns" About New Domains. *AFIPS National Computer Conference*. Anaheim, California: 39-46.

Ballard, B.; Lusth, J.; and Tinkham, N. 1984a A Transportable, Knowledge-Based Natural Language Processor for Office Environments. *ACM Transactions on Office Information Systems* 2(1): 1-25.

Ballard, B.; Lusth, J.; and Tinkham, N. 1984b An English-Language Processor for Office Environments. *AFIPS National Computer Conference*. Las Vegas, Nevada.

Ballard, B. and Tinkham, N. 1984 A Phrase-Structured Grammatical Formalism for Transportable Natural Language Processing. *Computational Linguistics* (to appear).

Biermann, A. 1981 Natural Language Programming. *Proceedings of the NATO Advanced Study Institute on Automatic Program Construction*. Bonas, France.

Biermann, A. and Ballard, B. 1980 Toward Natural Language Computation. *American Journal of Computational Linguistics* 6(2): 71-86.

Biermann, A. and Ballard, B. 1983 A Natural Language Interactive Computer System. *Army Conference on Applications of Artificial Intelligence to Battlefield Information Management*. White Oak, Maryland: 135-143.

Biermann, A.; Ballard, B.; and Sigmon, A. 1983; An Experimental Study of Natural Language Programming. *Int. Journal of Man-Machine Studies* 18(1): 71-87.

Biermann, A.; Rodman, R.; Ballard, B.; Betancourt, T.; Bilbro, G.; Deas, H.; Fineman, L.; Fink, P.; Gilbert, K.; and Heidlage, F. 1983 Interactive Natural Language Problem Solving: A Pragmatic Approach. *Applied Natural Language Processing*. Santa Monica, California: 180-191.

Fink, P. 1983 Conditionals in a Natural Language System. Master's Thesis. Department of Computer Science, Duke University, Durham, North Carolina.

Geist, R.; Kraines, D.; and Fink, P. 1982 Natural Language Computation in a Linear Algebra Course. *National Educational Computer Conference*: 203-208.

Lusth, J. and Ballard, B. 1983 Knowledge Acquisition for a Natural Language Processor. *Conference on Artificial Intelligence*. Rochester, Michigan: 859-870.

Lusth, J. and Ballard, B. 1984 The Design of DOMINO: A Knowledge-Based Information Retrieval Processor for Office Environments. Technical Report CS-1984-2, Department of Computer Science, Duke University, Durham, North Carolina.

Sigmon, A. 1981 The Semantics of Looping Structures in Natural Language Computation. Master's Thesis. Department of Computer Science, Duke University, Durham, North Carolina.

Announcements

Nominations for 1985 ACL Slate

The Nominating Committee has submitted the following slate for consideration during the 22nd ACL Annual Meeting, to be held during COLING84:

President

Madeleine Bates, *Bolt Beranek and Newman*

Vice President

Ralph M. Weischedel, *University of Delaware*

Secretary-Treasurer

Donald E. Walker, *Bell Communications Research*

Executive Committee

1985-1987

Alan W. Biermann, *Duke University*

1985 (to complete Weischedel's unexpired term)

Richard I. Kittredge, *University of Montreal*

Nominating Committee (1985-1987)

Martha W. Evens, *Illinois Institute of Technology*

As usual, nominations will be accepted from the floor, provided that the person nominated has indicated willingness to service, if elected.

□ □ □

Workshop on Relational Models

Stanford University – 29 June 1984

09:00 LEXICAL DATABASES AS DYNAMIC SYSTEMS OF REPRESENTATION

Nicoletta Calzolari, *University of Pisa*

09:30 A LEXICON FOR A STROKE DATABASE

Thomas Ahlswede and Martha Evens, *Illinois Institute of Technology*

10:00 DETERMINATION OF LEXICAL-SEMANTIC RELATIONS FOR MULTILINGUAL TERMINOLOGY STRUCTURES

John White, *Siemens*

10:30 BREAK

11:00 LEXICAL, SYNTACTIC, AND SEMANTIC ACQUISITION IN A TRANSPORTABLE NATURAL LANGUAGE PROCESSOR

Bruce Ballard, *Duke University*

11:30 IMPROVED RETRIEVAL USING A RELATIONAL THESAURUS FOR AUTOMATIC EXPANSION OF BOOLEAN LOGIC QUERIES

Edward Fox, *Virginia Polytechnic Institute*

12:00 THE BASELINE UNDERSTANDING MODEL (BUM)

Burghard Rieger, *Technical University of Aachen*

12:30 LUNCH

02:00 HOW TO TEACH A NETWORK

Oswald Werner, *Northwestern University*

02:30 THE HIERARCHICAL SCHEMA OF A DAILY NEWSPAPER

Alexander Nakhimovskiy, *SUNY Oswego*

03:00 FACTORING A KNOWLEDGE BASE

John Sowa, *IBM*

03:30 BREAK

04:00 COLLOCATIONAL RELATIONS IN A MEDICAL SUBLANGUAGE

Raoul Smith, *Northeastern University*

04:30 EXTENSIONS OF LEXICAL COHESION: SYSTEMATIC, INSTANTIAL, AND FIELD-BOUND RELATIONS IN TEXTS ABOUT LITERATURE

Mary Ann Eiler, *American Medical Association*

05:00 AN EXPLORATION OF GRADED SET MEMBERSHIP

Judith Markowitz, *Erikson Institute*

05:30 RELATIONAL MODELS AND THE PHILOSOPHY OF SCIENCE

William Frawley, *University of Delaware*