

ing morphosyntactic properties were analyzed within each language first and compared among the four languages:

- 1) the category of gender,
- 2) the category of animateness,
- 3) the category of case and number.

The product of this comparative analysis is a set of formation rules which embody a system for the identification of the inflected forms. The detailed result will be presented in an additional report.

Types of Language Hierarchy

E. D. Pendergraft

Linguistics Research Center, The University of Texas

Various relations lead to hierarchical systems of linguistic description. This paper considers briefly a typology of descriptive metalanguages based on such relations and sketches possible consequences for computational linguistics.

Its scope is accordingly limited to metalanguages having operational interpretations which specify individual linguistic processes and structural interpretations which specify language data of individual languages. Immediate-constituent, context-free metalanguages are used to illustrate hierarchical types.

Path Economization in Exhaustive Left-to-right Syntactic Analysis

Warren J. Plath

Computation Laboratory, Harvard University

In exhaustive left-to-right syntactic analysis using the predictive approach, each path of syntactic connection which originates at the beginning of a sentence must be followed until it is clear whether or not it will lead to the production of a well-formed analysis. The original scheme of following each path until it terminates either in an analysis or in a grammatical inconsistency has been considerably improved through the incorporation of two path-testing techniques. Using the first technique, the program abandons a path as unproductive whenever a situation is detected where the prediction pool contains more predictions of a given type than can possibly be fulfilled by the remaining words in the sentence. Employment of the second technique, which is based on periodic comparison of the current prediction pool with pools formed on earlier productive paths, eliminates repeated analysis of identical right-hand segments which belong to distinct paths.

Taken together, the two path-testing procedures frequently enable the program to terminate the processing of a path well before its end has been reached. For most sentences, this means a considerable reduction in the total path length traversed, accompanied by a corresponding increase in the speed of analysis. Comparison of runs performed using both versions of the program indicates that employment of the new techniques

reduces the average running time per sentence to less than one-fifth of its former value.

A Computer Representation for Semantic Information

Bertram Raphael

Computation Center, Massachusetts Institute of Technology

This paper deals with the problem of representing in a useful form, within a digital computer, the information content of statements in natural language. The model proposed consists of words and list-structure associations between words. Statements in simple English are thought of as describing relations between objects in the real world. Sentences are analyzed by matching them against members of a list of formats, each of which determines a unique relation. These relations are stored on description-lists associated with those words which denote objects (or sets of objects). A LISP computer program uses this model in the context of a simple question-answering system. Functions are provided which may grow, search, and modify this model. Formats and functions dealing with set-relations, part-whole and numeric relations, and left-to-right spatial relations have been included in the system, which is being expanded to handle other types of relations. All functions which operate on the model report information concerning their actions to the programmer, so that the applicability and limitations of this kind of model may more easily be evaluated.

Specifications for Generative Grammars Used in Language Data Processing

Robert Tabory

IBM Thomas J. Watson Research Center

It becomes more and more evident that successful pragmatics (i.e. automatic recognition and production procedures for sentences) cannot be performed without previously written generative grammars for the languages involved, using an underlying meta-theoretical framework proposed by the present school of mathematical linguistics. Two aspects of grammar writing are examined:

1. A taxonomy over the non-terminal vocabulary, using a subscribing system for signs and fitting into the more general string taxonomy of phrase structure components. The resulting more complex lexical organization is studied.

2. A command syntax for phrase structure components limiting the full, not necessarily needed generative power of these grammars. The proposed restrictions correspond to a priori linguistic intuition. Applicational order and location of the rules is studied.

Finally, the recognitional power and generative capacity of a computer are examined, the machine being structured according to a Newell-Shaw-Simon list system. It is well known that pushdown stores are particular cases of list structures, that context-free grammars