SOME APPLICATIONS OF COMPUTERS IN LINGUISTICS*

Victor H. Yngve

Computers have already had a considerable impact on linguistics, and there is every reason to believe that the impact will be far greater and far more important in the future. The application of computer methods in linguistics has been along several lines: There have been applications to traditional or usual linguistic methods. There have been applications to extensions of traditional methods made possible by the special abilities of the computer. These are both important, but most important of all are applications to methods that are entirely new to linguistics and that hold the exciting promise of new and deeper insights into language phenomena.

Computer applications to traditional or usual linguistic methods are the most straightforward. These are methods that have been useful in an unautomated form for many years. Concordance making, text searching, and the handling and sorting of linguistic data lend themselves to easy automation. The use of the computer can bring speed, convenience, accuracy, and relief from a certain amount of drudgery.

Computer applications that involve straightforward extensions of the older techniques hold the promise of yielding results virtually unattainable using the older techniques. This is because the superior speed, accuracy and clerical manipulating ability of the computer bring a new dimension to the research capabilities of the investigator. The kinds of operations envisaged in these extensions of older techniques would be entirely impractical without the computer because of the large amount of manual labor that would be entailed. The computer thus becomes an instrument for increasing or extending the scope and usefulness of older techniques into areas that had previously been effectively closed to investigation.

But some of the computer applications to linguistics are entirely new and are not just straightforward applications or extensions of older non-computer methods. The computer is opening up exciting new vistas in linguistic research. It offers opportunities for the exploration of virgin territory and the possibility of obtaining new and deeper insights into language, its structure and its use by the human organism. The full scope of the future possibilities is only dimly seen, but the results already in tend to indicate that the future will be very bright indeed.

It is thus important to understand these new techniques and to develop them and apply them systematically in linguistics. A representative and diverse selection of

* This work was supported in part by the National Science Foundation.

applications is discussed here. We do not here present a complete survey:[1] there are other applications that are perhaps equally important. But the ones presented here should serve to indicate the kind of an impact that computers are already having in linguistic research. In this new area, progress may be limited only by the bounds of our creative imagination.

File Processing

Many existing linguistic procedures involve the handling of large files of data or large quantities of text. In each case the application of the computer brings certain advantages. In the case of files of data, computer handling makes possible frequent updating involving the interfiling of new material, and then, with each updating, the whole file can be printed out in its new form. Thus the linguist always has available the whole current file arranged and printed in a convenient format. With the file stored in a form amenable to computer operations, the possibility is open for easy production of complex new arrangings and sortings, with printed copies in each new arrangement. Or it is possible to make special searches of the file for particular types of items answering to certain specified search criteria. The ease with which specialized searches of the data can be carried out makes possible a considerable flexibility in research, for new searches can be planned on the results obtained from previous searches.

The advantages that automation brings to these rather straightforward file operations are several. First of all there is the advantage of accuracy in sorting and copying. The accuracy of the computer far exceeds that of manual sorting and copying. Another advantage is the more flexible and convenient arranging and displaying of the data without the necessity for extensive manual operations. Then there are the speed and flexibility in research that are gained by having the data in a tractable form where they can be easily searched, sorted, arranged and printed.

But perhaps one of the greatest advantages of the automation of file operations is the possibility of introducing sophisticated error checks and controls based on known regularities in the data entries. For example, in a large file of personal names in the ancient Semitic Amorite language,[2] it is presumed that the names will conform to a certain known internal morphological structure. It is thus possible to program computer checks of the accuracy of the manual copying and transliteration based on the presumed structure of the entries. Any names that do not conform to the posited structure are automatically located and marked by the computer so that they can be examined in detail to determine whether the deviation is significant or whether it is due to an inadvertent error in copying or keypunching the data. By such means the data can easily be maintained in a state of accuracy much higher than is generally feasible without an inordinate amount of manual checking.

Text Handling

In the case of text handling, there are several operations that become quite easy with the availability of text in machinable form. The first thing that comes to mind, of course, is the preparation of concordances, a task that has been a very timeconsuming one for scholars in the past. With the use of the computer and existing concordance programs, concordances can be obtained relatively easily.

Related to the production of concordances are a number of other operations that can easily be automated. These include the extraction of vocabulary from text and the counting of words, morphemes, or other items of interest. These operations and other more sophisticated ones are being extensively applied in dictionary and glossary making, as well as in areas of literary and textual criticism and stylistics.

With the availability of text in the computer, certain important extensions of concordance techniques emerge. These involve the ability of the computer to carry out searches of the text according to complex search criteria. The limitations of the concordance that this possibility overcomes are twofold. In the first place, the typical concordance arranges segments of the text alphabetically according to each of the words of the text, so that the investigator can look up any word and find all of its contexts brought together. But typically the investigator has neither the interest nor the time to look up every word in the concordance. But also, typically, he could not use a partial or selective concordance because he cannot foresee which words he is going to want to look up, because he cannot foresee the exact course that his research will take. The ability to carry out text searches to order overcomes this difficulty, for a number of searches can be carried out during the course of the research.

The second difficulty of the traditional concordance is that, although it is easy to find all of the contexts of a given word, it is difficult to use a concordance to help find examples of more complex patterns, for example, sentences involving inversion, sentences involving three or more clauses, sentences involving one of a number of negative adverbs and a progressive verb form. Searches of text for patterns such as these become possible with the newer computer techniques. Particularly important in achieving the flexibility and ease of programming required for specifying and carrying out such complex searches is the use of a convenient high-level programming language. The flexibility of the clerical tasks that can be accomplished by the computer to order, especially if a high-level programming language like COMIT[3] is employed, is so great that the effect of this approach is to give the investigator a much more powerful tool for searching his text than he could possibly expect by using concordances.

Bringing Structure to Light

The use of the computer with its increased speed and flexibility opens up the possibility of entirely new operations on text and on data that promise to add new dimensions to Linguistic research. Some of the new possibilities involve the use of statistical techniques for revealing suspected structures in a text. These methods include some of the cryptanalytic techniques that have been used so successfully in deciphering texts in unknown languages. But the possibilities are much broader than appear to be realized, for the methods have not been generally explored or exploited to the extent that they deserve.

The basic idea behind some of these methods is that there is an antithesis between structure and randomness. Randomness is lack of structure: A completely unstructured sequence of characters would exhibit all the properties of randomness that are known to mathematicians, but if the sequence of characters is structured in any way, the randomness is in some measure destroyed. Therefore, if statistical measurements are made on the text and deviations from randomness are found, the deviations are to be attributed to the influence of structure. The technique thus involves a search for ways in which the text deviates from randomness and requires the finding of statistical measures that are sensitive to the constraints of interest to the investigator.

Now at any given point in the course of research, a certain amount of the structure of the text is understood and describable, and the remainder of the structure is not understood and is undescribed. The statistical techniques are to be applied to the problem of discovering some of this unknown structure. It is thus necessary, when setting up the statistical tests, to cancel out the known and measurable effect of the known structure. This can sometimes be done in a rather straightforward manner; sometimes it requires considerable ingenuity.

An early experiment along these lines dealt with a scheme called gap analysis,[4] which was pursued as a method of looking for syntactic constraints. The experiment, using an English text of about ten thousand words, was aimed at exploring the possibilities of the technique. It was assumed that the known and measurable structure was the different frequencies of different words in the text. The statistical structure of the text was then examined to look for deviations from randomness arising from the constraints of syntax. If there were no constraints due to syntax, the occurrence of a word in a text would have no effect on the possibilities or probabilities of occurrence of other words in the vicinity. The experiment, however, showed up strong deviations from randomness, and these could be used to posit syntactic constraints that agreed with what is known about English syntax. The method was thus shown to be capable of giving results.

Dialect Survey

There are many other possibilities opened up by computer techniques. Let us consider the following possible application to dialectology that could assist in a large-scale survey of the phonemic structure of related dialects. A modified informant technique would be used that would not require the presence of a trained linguist. This, together with the use of the computer for data reduction, would make possible the gathering and processing of phonemic dialect information on an unprecedented scale.

A linguist with some knowledge of the dialects in question would prepare a list of words to be presented to the informants. For each word, a punched card would be prepared which contained the word and a sentence exemplifying its use. The word and sentence would appear printed along the top of the card. The cards would be made up into decks and distributed to the informants together with carefully worked out instructions. The informants would be instructed to sort the cards into piles according to criteria designed to reveal phonemic similarity and difference. The informant would then check over the piles that he has made and add to the top of each pile one of the heading cards provided. The various piles of cards, separated by their heading cards, would then be returned to the linguist who would have a computer program ready for reading the cards and summarizing the information implicit in the sorting done by the informants.

In order to make use of large amounts of data of this type, programs would have to be developed for grouping and summarizing the data according to techniques of set theory or statistics. These operations can all be applied automatically. The result would be a rather complete catalog of informant responses arranged according to dialect similarity. Compact statements of similarities and differences would then be made on the basis of the processed data, and these could be used as a basis for more thorough and detailed investigations in the field. In fact, the availability of summarized data from a large number of individuals would allow the linguist to select the best small set of informants for further detailed phonetic and phonemic investigations by personal interview. The technique would thus allow a choice of informants that would reduce the over-all effort involved in obtaining the desired dialect descriptions.

Testing of Linguistic Statements

No linguistic statement or description can be trusted unless it has been adequately tested against the linguistic data that it is supposed to cover. The difficulties of comparing a statement with data become very great as the complexity and wealth of detail covered by the statement increase. In this area, computer techniques are finding a very important application. The utility of computer methods in testing is particularly evident in generative morphology and syntax. However, this is not the only area where computer techniques are relevant. Paradigmatic descriptions, historical statements and all other methods of linguistic description that are sufficiently explicit and precise are amenable to computer testing.

Linguistic statements can be tested in two ways, by synthesis and by analysis. The testing by synthesis may involve synthesizing all forms that are predicted, as in the case of the author's statement of the inflection of the English regular and irregular verbs.[5] A program to test this statement synthesized and printed a complete paradigm for each of the irregular verbs and for representative regular verbs.[6]

But if the linguistic statement involves more variability, as in the area of syntax, it may be difficult or impossible to synthesize and print out each of the described sentences. In fact, most syntactic statements generate an infinite set of sentences, and it is in principle impossible to synthesize and print them all. A method of random generation thus has much to recommend it.[7-10] According to this scheme, sentences conforming to the grammatical constraints expressed in the grammar are synthesized at random. In the generating of a sentence, a random choice is made at any point where there are alternative constructions that could fulfill a given function. The resulting generated sentences can then be examined and compared with observation. If the generated structures do not conform to the language, the linguist knows that his statement is inaccurate in certain respects. He is thus enabled to make the appropriate corrections.

In the testing of generative grammars by analysis, data in the form of texts are fed into the computer and analyzed according to the grammar. The results of the analysis are then compared with what the linguist expects, and any deviations again lead to improvements in the linguistic statements.

The two methods of testing a statement, synthesis and analysis, are complementary in a certain sense. The method of synthesis tests to see that the structures conforming to the linguistic statements are legitimate expressions in the language. The method of analysis tests to see that expressions known to be in the language are actually covered in the statements.[11, 12]

The fact that our statements are currently only a partial explanation of language phenomena shows up clearly in such testing. In the case of synthesis of sentences from a generative grammar, the sentences may be syntactically acceptable but nonsense. Some of the difficulties of comparing generative grammars with observation are concerned with judging the grammatical acceptability of nonsense. This difficulty is a natural consequence of the fact that generative grammars do not deal with the distinction between sense and nonsense. Perhaps it makes no sense to ask whether nonsense is grammatical or not. In the case of the analysis of sentences by computer, the fact that the linguistic statement is only a partial explanation shows up in the extreme degree to which the parsings produced involve multiple syntactic ambiguity.[13]

Models of Language Users

If a computer program can analyze and synthesize sentences according to a linguistic statement, the program itself can be considered a theory in the sense that it makes predictions. Thus as our knowledge advances of how to test grammars by means of programs, we may find that it will be reasonable to make no distinction between the program and the linguistic statement. This practice becomes quite feasible with the use of high-level programming languages such as COMIT,[3] by means of which a computer program can be written in a way that is convenient for the linguist to read and comprehend.

But the most exciting implication of computers to linguistics follows from the fact that both man and computer are symbol manipulators or information processors. For this reason, a computer simulation of linguistic behavior stands a chance of giving us much deeper insights into language phenomena than computer simulation might provide in other areas, such as the simulation of traffic flow in a city or of material flow in a manufacturing process. In other words, a computer program may be a model of man in his role of symbol manipulator in a much deeper sense than a computer program may be a model of other processes, because the computer is also a symbol manipulator.

An example of the heuristic value for linguistics that a computer model of language behavior can provide is to be found in the work on the relation of the temporary memory to linguistic structure.[14,15] In this work, a computer program was devised to model a certain facet of human language behavior, namely the production of grammatical sentences. This led to a more unified understanding of a wide diversity of previously unconnected facts of the structure of English, and led to a unified view of syntax that promises to be extremely important in the understanding of language typology and language change. On the basis of this work it has become possible to comprehend perhaps the major reason for the complexity of languages.

It may be safe to say that we will only really understand human language behavior when we can make working models that also exhibit language behavior. The emergence of the computer as a tool in linguistics puts at our disposal the very techniques that we need for making such working models, and the prospect is extremely exciting.

NOTES

1. There are no adequate surveys, since the field is new and advancing very rapidly. The reader may find additional material in the following references: *Mechanical Translation and Computational Linguistics,* a quarterly published by the University of Chicago Press for the Association for Machine Translation and Computational Linguistics, H. H. Josselson, Sec., Wayne State University, Detroit, Michigan; *The Finite String,* a newsletter published for the above organization—each issue includes bibliographical information; H. P. Edmundson (ed.), *Proceedings of the National Symposium on Machine Translation* (Englewood Cliffs, N. J.: Prentice-Hall, 1961); P. L. Garvin (ed.), *Natural Language and the Computer* (New York: McGraw-Hill, 1963); D. G. Hays (ed.), *Readings in Automatic Language Processing* (New York: American Elsevier, 1966); S. M. Lamb, “The Digital Computer as an Aid in Linguistics,” *Language,* XXXVII, No. 3 (1961), 382-412; W. P. Lehmann, “Zwischen zwei Sprachen,” *Graduate Journal* (University of Texas), VII, No. 1 (1965), 111-131; A. G. Oettinger, *Automatic Language Translation* (Cambridge, Mass.: Harvard University Press, 1960).

2. I. J. Gelb, “On the Morpheme *ān* in the Amorite Language,” above (p. 45).

3. V. H. Yngve, *COMIT Programming* (Cambridge, Mass.: M.I.T. Press) (in preparation).

4. V. H. Yngve, *Gap Analysis and Syntax* (“IRE Transactions on Information Theory”), Vol. IT-2, No. 3 (1956), pp. 106-112.

5. V. H. Yngve, “MT at M.I.T., 1965,” in A. D, Booth (ed.), *Machine Translation* (Amsterdam: North-Holland Publishing Company, 1967).

6. A number of others have also used computer methods to test morphological statements. For example, Erica Reiner and James McCawley, both of the University of Chicago, have used the computer to test generative statements of Akkadian verb morphology and of Finnish verb and noun morphology (private communications).

7. V. H. Yngve, “Random Generation of English Sentences,” *1961 International Conference on Machine Translation and Applied Language Analysis,* I (London: Her Majesty’s Stationery Office, 1962), 66-80.

8. A. C. Satterthwait, “Computational Research in Arabic,” *Mechanical Translation,* VII, No. 2 (1963), 62-70.

9. D. A. Dinneen, “The Grammar of Specifiers,” in D. G. Hays (ed.), *Readings in Automatic Language Processing* (New York: American Elsevier, 1966), chap. 8, pp. 127-136.

10. G. H. Harman, “Generative Grammars without Transformation Rules, a Defense of Phrase Structure,” *Language,* XXXIX, No. 4 (1963), 597-616.

11. The author and his students are continuing to develop a computer grammar of English that can be used for sentence synthesis and analysis. The current scheme involves a phrase-structure grammar with discontinuous constituents and subscripts.

12. Work is also proceeding on computer grammars within the transformational-generative framework or some modification of it at the MITRE Corporation, Lexington, Massachusetts; under Zellig Harris at the University of Pennsylvania, Philadelphia, Pennsylvania; and elsewhere. Unfortunately computer methods are not being widely applied by transformational grammarians, and many transformational descriptions remain untested. It seems to be little realized that methods of computer-program debugging (error discovery and elimination) can be applied profitably in linguistic research.

13. See the more recent literature in the field of mechanical translation.

14. V. H. Yngve, “A Model and an Hypothesis for Language Structure,” *Proceedings of the American Philosophical Society,* CIV, No. 5 (1960), 444-466.

15. V. H. Yngve, “The Depth Hypothesis,” in *Proceedings of Symposia in Applied Mathematics* (“Structure of Language and Its Mathematical Aspects,” Vol. XII [American Mathematical Society, 1961]), pp. 130-138.