THE IMPACT OF ON-SITE COMPUTING ON FIELD LINGUISTICS: THE POWER OF MAN AND MACHINE IN INTERACTION

Gary F. Simons

0. INTRODUCTION

The microelectronic revolution is drastically changing not only the way we live, but also the way we do linguistics. On-site computing is bringing about a5 qualitative change in field linguistics, not just a quantitative change. In addition to speeding up certain analysis steps, computing power is improving the quality of work by opening up methods of analysis made practical only by the interaction of man and machine. The essential nature of this interaction is that the linguist, who excels in formulating hypotheses and making analytical decisions, uses the computer, which excels in the tedious tasks of filing, searching, sorting, counting, and comparing, as a tool for testing the hypotheses he formulates. The results of such tests allow the linguist to sharpen his hypotheses or begin again with new ones. With this division of labor, the computer is doing what it does best, tedium, and the linguist is free to do what he does best, thinking. This cycle of interaction continues until a satisfactory solution is reached.

This paper is written in three sections. Section 1 reviews the history of the microelectronic revolution and shows how on-site computing for the field linguist has been made possible. Section 2 turns to specific applications of computer technology in field linguistics, giving an overview of the field linguist's task and how computers can fit in and answering some popular misconceptions concerning computers in linguistics. Finally, Section 3 develops the main thesis that on-site computing is bringing about a qualitative change in field linguistics.

1. THE MICROELECTRONIC REVOLUTION

The microelectronic revolution broke forth in the 1960's, largely as a result of the United States' effort to land a man on the moon. This project required that very sophisticated electronic equipment for communication, navigation, scientific experimentation, and computation be placed on board a small spacecraft. In the early 1960's it would never have fit, but by 1969 microelectronics had squeezed tremendous computing power into the Apollo spacecraft and the mission was successful.

Since the moon landing, the pace of development has not slowed down. The miniaturization of electronic circuits has increased at phenomenal rates. At the same time, the cost of circuitry drops dramatically every year. Today we see the spin off of the microelectronic revolution affecting us in almost every aspect of our daily lives. The following is a list of some products which now incorporate microelectronic controllers or computers of various degrees of sophistication: children's toys, video games, pinball machines, vending machines, digital watches, sewing machines, microwave ovens, washing machines, traffic signals, automobile engines, radio tuners, televisions, office typewriters, handheld calculators, home computers--and the list could go on. Of special interest to the field linguist is the affordable desk-top computer that has made on-site computing a reality.

To give an idea of the incredible advances that have been made thus far and to give you a glimpse of what the future may hold, I will back up in time and trace electronic developments up to and beyond the present. Computers work in a binary (or base two) number system. The only digits in the binary number system are zero and one. These binary digits are referred to as "bits". In electronic implementation of the binary system, the two digits are represented by distinct electronic states such as a switch either on or off, a voltage either high or low.

In the earliest electronic computing machines of the late 1800's and up through the middle 1900's, mechanical switches of various types were used. The most popular was the relay, a switch that can be turned on or off automatically by an electromagnet.

In the 1940's and 1950's the vacuum tube replaced the relay as the basic switch. (See Figure 1 for a scale drawing of this and other components). The vacuum tube is a purely electronic switch with nothing mechanical. Thus without the latency times of moving parts, the tube made a faster computer. The first large electronic computer, the ENIAC, began operating in 1946 and was made up of 18,000 tubes. As for size it filled a 30' by 50' room and consumed 130,000 watts of power.

The tube offered a number of disadvantages, however. It was bulky, consumed much power, and had to be replaced often because the filaments burned out. The semiconductor transistor, invented in 1949, provided the solution for all these problems. It performed an equivalent electronic function but was very small, very low power, and very rugged by comparison. (See Figure 1.) The transistor, and the microelectronic integrated circuits which have followed it, are basically chips of silicon (which happens to be the most plentiful element in the earth's crust) with impurities etched in patterns which define the electronic components and provide paths for the flow of electrons.

In 1959, the integrated circuit was developed. The integrated circuit operates on the same principle as the transistor. It, however, contains many transistors (along with resistors, capacitors, and diodes) etched into a complete electronic circuit, all on a single chip of silicon.(See Figure 1.)

Since the introduction of the integrated circuit, microelectronic technology has advanced at an incredible rate. In 1959, the state of the art was such that one transistor could be etched on a silicon chip. Every year since 1959, the number of electronic components (transistors, resistors, capacitors, diodes) that could be etched on a chip approximately 1/4 inch square has doubled (Noyce 1977). Figure 2 is a graph which plots this phenomenal growth. Note in reading the graph, that the vertical axis is on a logarithmic scale. On a normal linear scale, this graph would be an exponential curve, going nearly straight up in later years. On the logarithmic scale the curve is straightened out so that we can more easily study the trend.

In reading the graph in Figure 2, we see that in 1962, the year the Congress made the commitment to fund the project for putting a man on the moon before the end of the decade, the density of integrated circuits was eight components per chip. By 1969, the year the expedition actually landed, the density was 1024 components per chip. That is, the electronic circuitry was 128 times more dense. Also it would have consumed about 128th as much power, because power consumption is related more to the physical size of the chips than to the number of components.

This incredible growth did not end when the Apollo project ended. It has continued to the present so that in 1977, the point at which the graph (Figure 2) was written, the density of an electronic memory chip which can store 65,000 bits of information, is nearly 262,000 components. That means that computer circuits made from 1977 level technology will be 256 times more compact than equivalent circuitry that went to the moon. If we compare today's component density with that of the early and mid 1960's, we see that the desk-top microcomputers of today are as powerful as machines that were thousands of times larger only 15 years ago. And those early computers consumed about that many times more power and cost about that many times more to buy. The microcumputers of today are not toys; they are sophisticated computers in very small and very low cost packages.

As we project into the future, there is nothing to indicate that the doubling trend is slowing down. The technology is still far from the fundamental limits imposed by the physics of the silicon, the impurities, and the electrons (Noyce 1977:65). A deviation from exponential growth is inevitable, but it is not yet in sight.

As we enter the 1980's, microelectronic technologies more dense than integrated circuitry are becoming feasible. One such technology, called the bubble memory, stores bits of information in the polarity of micro-miniature bubbles of magnetism. Whereas the large-scale integrated circuit memories of the 1970's can store thousands of bits, the bubble memories of the 1980's will store millions of bits.(See Figure 1.)

Figure 3 plots the decline in cost of microelectronic circuitry. The graph shows that on average, the cost of computer memory has been decreasing about 30% every year. The costs of other microelectronic components in computers have followed similar trends. However, those parts of the computer which interface the microelectronic circuitry to the outside world--the keyboards, video displays, printers, and mass storage devices--are not falling in price as dramatically.

Figure 4 shows trends in the utilization of electronic functions. This graph shows the extent to which the world is embracing more and more microelectronic products every year. Note that the definition of electronic functions (in the caption to the graph) means that it increases both when a new product is purchased or when an old product is upgraded by putting denser circuitry inside of it. The curves predict that in ten years the world's use of electronic functions will be 100 times what it is now. Broken down to an annual rate, this means that every year 60% more functions will be embraced. As we see the trend which the world is following, this could serve as a measuring stick for our own organization. If the video game and pinball machine manufacturers are upgrading their operations and production by incorporating 60% more electronic functions every year, how much more should we who have an infinitely more meaningful task take advantage of this powerful technology at least to the same extent. One year from now, will we be utilizing 60% more electronic technology than we are today? In ten years, will we be using 100 times more electronic technology than we are today? The potential is there if we consider the kind of computer support outlined in the next section for all of our linguistic teams available on a daily basis.

2. COMPUTERS IN FIELD LINGUISTICS

Two aspects of computers in field linguistics are discussed in this section. First, there is an overview of the linguistic field project showing specific application areas where computers can help. Second, there is a discussion of general principles about what computers can and cannot do and how best we can use them in linguistics.

2.1 An Overview

An overview of the linguistic field project is given in Figure 5. Four kinds of labels are defined in the diagram. Those in all upper case letters represent basic data files, which the linguist collects. The labels in lower case letters represent the final results for which the linguist is aiming. The labels in dashed boxes represent the procedures in which a computer cannot be of any help. Those enclosed in solid boxes are the procedures in which computers can be of help. The arrows indicate the normal sequence of steps from data to analysis procedures to results and so on.

There are three things to keep in mind when looking at the diagram. (1) It is not complete; many more data files and procedures and results could be included. The diagram is meant to hit the major points. (2) It is oversimplified in that the computer assisted procedure steps are complex steps involving intermediate data files and human thinking steps as well as computer programs. (3) Paths in the diagram are traversed more than once. The analyst continues cycling through the paths year after year as more data are collected and analyses and descriptions are refined.

The discussion which follows talks about what it is possible for the computer to do in helping the field linguist. Nearly all of the programs mentioned have been written at some time for some computer. However, we are still a few years away from having all the programs described available in an integrated program package for the field linguist on a computer he can afford. This is currently a project of the Summer Institute of Linguistics, and progress is being made toward that goal.

There are three main data files involved in linguistic analysis. These are the LEXICON, TEXTS, and GRAMMAR. Computer data bases can replace the card files and notebooks with which linguists are all familiar. Initially, the data are collected by non-computational means as shown by the dotted boxes in which observation, elicitation, and comparison (as in comparison with related languages) are listed as data collection techniques.

Once the initial data are collected, computer aided analysis steps may begin. This is indicated by the arrows leaving the data files. In addition, some of the computer methods serve to complement the observation and elicitation methods as a means for acquiring more data as the project progresses. This is indicated by arrows going to the data files.

Beginning with the TEXT files, a computer can help to expand the LEXICON by looking up all the forms that occur in the text and compiling a list of all new forms for the linguist. After the linguist fills out the lexical entry for each one, these new entries can be automatically inserted into the lexicon. Text also serves as a source of example uses for inclusion in the lexicon.

The LEXICON provides the basic data file for phonological analysis. From the phonetic transcriptions in the LEXICON, programs can automatically tally phone occurrences and co-occurrences, and canonical shapes of syllables and words, and they can find all examples of forms that fit a given description (Grimes, Alsop, and Wares 1968; Simons 1977b). These computed results provide the basis on which the linguist can posit and test an analysis. The eventual result is a phonological description.

The phonological description leads to an orthography. After orthography decisions are made (or changed) a consistent changes program can be used to automatically change the spelling of the entries in the LEXICON, and the computer can then rearrange the file into alphabetical order.

The phonological description also serves as input to a program which automatically generates a list of all the phonologically possible words or stems in the language. This word list benefits the language analysis in at least two ways. (1) It enlarges the LEXICON by uncovering many previously unknown actually occurring forms. (2) It refines the phonological analysis, as it is found that some phonologically impossible forms are in the list or that the list does not contain some forms that are phonologically possible.

The LEXICON serves as input for programs to aid in lexicography. Some such programs can be for quality control to check completeness of entries and consistency in use of labels. Others can pull out entries of the same grammatical or semantic class for detailed study. When using a highly constrained, but explicit, model for lexicography such as the meaning-text model described by Grimes (1980), programs can be used to automatically generate all the cross-references implied by a given entry and to trace out definitions to ensure that they are not circular. All of these types of lexicographical procedures serve to improve the quality and increase the extent of the LEXICON data base. Another class of lexicography programs formats the LEXICON data base and converts it into a printed form suitable for publication as a dictionary.

The phonological description, dictionary, and grammatical description shown in Figure 5, as well as the anthropology description, primers, texts for reading materials, and description for dialect situation shown in Figure 6, all show a computer-assisted procedure called word processing. Word processing is a two-part system of programs that helps in the preparation of any report for circulation or publication. The first part is a text editor which allows the linguist to enter the text onto computer storage media such as tapes and then to refine successive drafts of the report by changing the text on the media. The second part is a text formatter which takes the text on computer storage media and automatically produces a typed or typeset copy of the report in a format suitable for distribution.

Another computer assisted procedure leading from the LEXICON, which helps the linguist toward the goal of language learning, is computer-assisted vocabulary drill. Note also that grammatical patterns or paradigms from the grammar file can be drilled in the same way. The linguist can use the computer to maintain a regular schedule of programmed review (Henderson 1975). If the linguist wants to learn a particular lexical or grammatical item, he makes the initial effort of learning it and then enters it into the computer's programmed review file. If the item is entered on day x, then the computer will automatically bring it up for review on days x + 1, x + 2, x + 4, x + 8, x + 16, x + 32, and so on. As long as the linguist gives the proper response, the interval until next review is doubled. If he forgets the item, however, it starts again at the beginning of the sequence as though it had just been entered. In this way the language learner can use the computer to maintain a regular daily pattern of programmed review.

Going back to the TEXTS file in Figure 5 and following the path down to the right side of the diagram takes us into the area of grammatical analysis. The data file for the GRAMMAR is envisioned as consisting of two separate files. The first is a data base file of grammatical observations; it is like a conventional grammar notebook with the observations filed away under subject categories. The second file is a formal grammar; it is a precise grammatical description in terms of formulas or networks which the computer can use to parse (or analyze) forms in the language.

The TEXTS are the basic data file on which the grammatical analysis is based. Leading from TEXTS to GRAMMAR is a group of computer-assisted procedures. The first, labelled "discovery procedures", is a partial automation of Robert Longacre's <u>Grammar Discovery Procedures</u> (1964). The linguist can use an editor-like program to bracket all of the constructions which he wants to study and to enter a first etic analysis for each. Then a program can automatically extract all of the constructions from the texts and organize them into printed workcharts which bring together all examples sharing the same etic analysis. From there the procedure consists of refining the analysis of the individual data and having the computer generate new charts which reflect the reanalysis. This continues until the linguist is satisfied with the result.

The second procedure, labelled "concordance", is already familiar to field linguists. This involves the automatic compilation by computer of concordances of text material. With a large computer installation, this is best done by preparing a voluminous printout of every word and all its occurrences. With a small computer installation this may best be done by specifying the particular word or words to be analyzed, and then having the computer extract from the texts a concordance listing for those words only.

The third procedure, labelled "co-occurrences", is a set of programs that help the linguist to sort out the relations among co-occurring morphemes. Positional analysis helps sort the morphemes into position classes and establish their relative ordering (Grimes 1967). Restriction analysis helps establish the co-occurrence restrictions that hold between members of closed systems (Grimes, Lowe, and Dooley 1978). These latter three computational procedures lead the linguist from the TEXTS to observations and formulas for inclusion in the GRAMMAR file.

A method in grammatical analysis that parallels the generated blank word list in lexicography and phonological analysis is the automatic generation of paradigms and syntagms (constructions). For instance, a program can generate all the possible forms derived from a newly found verb root, assuming that it follows regular rules of formation. When checking the generated paradigm with an informant, discrepancies will indicate one of two things: (1) an irregular form of the verb which requires special treatment in the lexicon, or (2) an error in the linguist's rules for deriving the regular forms of a verb. Where a language has word classes based on patterns of word formation or agreement (such as a gender system), a computer can generate paradigms for a list of new words as though they were in each word class, then an informant can select the correct paradigm for each word. If the linguist has to be involved in this checking process, then it would probably not offer significant advantage over the conventional elicitation methods. However, if the informant by himself could interact with the program while the linguist was doing something else, then it could be a significant advance, especially if the program prepared a typed report for the linguist at the end!

Leading from the GRAMMAR file to a grammatical description is a procedure called data base manipulation. If the grammar file is organized like a data base, with observations tagged by subject categories, then data base manipulations such as extracting lists of related observations, reorganizing the observations, and pulling out relevant examples will be useful when preparing a grammatical description.

The final computer assisted procedure in Figure 5, grammar testing, links all three of the basic data files. Given a GRAMMAR (in this case the formal grammar) in computer readable form and a LEXICON in computer readable form, a grammar testing program can take a sample of TEXT, look up all the forms in the lexicon to find their grammatical categories, and then use the formal grammar to parse (or analyze) the structure of the text. If the program can successfully and correctly parse the text, then the lexicon and the grammar are correct as far as the given data are concerned. If the parse is not successful, then the linguist knows that either the lexicon or the grammar is incorrect. The grammar testing program will produce diagnostic output that indicates exactly where the parse failed and thus point directly to the problem in the lexicon or grammar. Automatic grammar testing will introduce a degree of objectivity and thoroughness previously unknown in grammatical analysis. Network Grammars, edited by Joseph Grimes (1974), gives a good introduction to the field of automatic grammar testing.

The field linguist often finds himself working in related disciplines. Figure 6 gives an overview of some ways computer assistance can be applied in anthropology, literacy, and language survey. Note that the three basic data files from linguistic analysis--TEXTS, LEXICON, and GRAMMAR--carry over into these related disciplines.

Anthropological observations can be kept in an ANTHROPOLOGY FILE. This file is a data base with observations filed under subject codes (such as those in Murdock's Outline of Culture,1950). Cultural items from the TEXTS and LEXICON can be copied into the ANTHROPOLOGY FILE. Observation and elicitation are other sources of data. Data base manipulations are used to extract and organize the information in the file to produce anthropological descriptions.

In the field of literacy and literature production, the TEXTS can be formatted directly into literature for publication. For primer construction, the GRAMMAR and TEXTS serve as the data source for a computational analysis of functors. The LEXICON and TEXTS serve as the data source for a computational analysis of grapheme and syllable shape productivity. The results of these two analyses allow the linguist to posit a sequence for teaching the graphemes, syllable shapes, and functors in the primer. Then a program automatically extracts from the LEXICON a list of all the words available for inclusion in the primer lessons at each step in the sequence. Another method of writing primer stories, that of simplifying existing texts by modifying all occurrences of untaught items, can also be facilitated by computer help. The computer can analyze texts to indicate what types of simplification a given text is most amenable to, and to mark all words which must be modified. Once a primer is written, a computer program for consistency checking can be used to check that untaught items do not occur in lessons, that new built words are not introduced at too great a rate, and that new words are sufficiently reviewed in following lessons.

A language survey produces a COMPARATIVE WORDLIST data file. This single data file can be submitted to a number of different computational analysis techniques for both the quantitative and qualitative assessment of dialect differences. Methods which have already been programmed include lexicostatistics, lexical isogloss analysis, phonostatistics, comparative method, phonological isogloss analysis, and refined phonostatistics (Simons 1977a). These analyses lead to further computational methods for the optimal allocation of personnel and resources in vernacular literature programs (Grimes 1974; Simons 1979) and for computer-assisted dialect adaptation of literature (Weber and Mann 1979).

2.2 Some Popular Misconceptions

Having surveyed the specifics of how computers can fit into the field linguistic project, I now want to give an overview of some general principles about what computers can and cannot do and how we can best use them in field linguistics. I want to deal with these in terms of seven popular misconceptions that I have heard expressed about computers.

(1) Computers can do anything; they can solve any problem.

A computer's brain is nothing more than a lifeless collection of silicon chips. It can do nothing until its human master tells it what to do and exactly how to do it. If a linguist cannot make a particular analysis decision, then certainly his computer cannot, for the computer depends on the linguist to tell it how to make the decision. The difference between computers and humans is that while humans are smarter, computers are faster and more consistent.

(2) Computers will take the human, creative element out of our work.

For the reasons discussed already under (1) above, this is not true. The computer is basically a highly efficient secretary. While the human thinks and creates, the computer takes care of the drudgery and tedium in analysis.

(3) We will have to re-learn analysis techniques to use a computer.

As already stated in (1) above, the computer can't do anything unless a linguist tells it exactly how to do it. For this reason, the analysis techniques the computer will use are precisely some of the ones we are already using by hand.

(4) We will have to become computer programmers to use a computer.

Here the burden falls on the professional programmer. Programs should be and are being written such that the naive user need know nothing about programming or the internal details of a program. He need only know the format of the input data, how to respond to program queries, and the format of the results. As far as the naive user is concerned, the computer can be viewed as little more than an intelligent typewriter. For this to work, the programmer must prepare thorough instructions for the user.

(5) By using computers we will become so dependent on them that we will be helpless when they break down.

Again the burden falls on the programmer. The computational methods should provide backup in the form of the conventional methods. That is, analysis steps should produce printed output that the linguist can put in a notebook or card file (just as though he had done it by hand). For instance, whenever the lexicon file is updated, filing slips can be printed out (including multiple copies in all the needed permutations) and inserted into a conventional card file kept for back up. Periodical printouts of the complete lexicon in book form provide an even more usable back up. With a good back up approach, there is no waste involved when the computer fails altogether. The linguist's data and analysis files are in much better shape than if they had been done only by hand, and the linguist is in a position to continue by hand.

(6) Computers will make our work go twice as fast.

It will still take just as long to learn a language adequately, even if computers are used. Analysis and translation will still be long and tedious tasks. Although computers are likely to save a few years on an otherwise fifteen year project, the difference isn't likely to be anything so dramatic as half the time. The dramatic difference will lie in the quality of the work. Within the same time frame, a computer assisted project will produce a publishable dictionary as a natural by-product, while the manual project will still have the lexicon in a rough card file or flat file. The computer assisted project will have tackled many more grammatical analysis problems than the manual project could in the same amount of time. The computer assisted project will have objective assessments of how accurate phonological and grammatical analyses are, while the manual project can only give a subjective guess. The computer assisted project can change orthographies or analyses or descriptions with ease, while the manual project is more likely to avoid change. The computer assisted project, with the integration of data files, analysis procedures, and word processing, is likely to produce more publishable results than the manual project.

(7) 80% or 90% of what a linguist wants to do with a computer is text editing.

I have heard this said by non-linguist programmers, but I do not agree. I'm convinced that the true percentage will be well under 50%. With the advent of personal, on-site microcomputers we are seeing a qualitative change in computing techniques from batch processes to interactive ones. No longer does the linguist spend his time editing a large data file so that he can hand it off to a big program that will hand back a lengthy printout. Now the linguist can use the computer as a daily tool in analysis work, interacting with the machine as hypotheses are formulated, tested, refined, and so on.

3. A QUALITATIVE CHANGE IN FIELD LINGUISTICS

On-site computing is bringing about a qualitative change in field linguistics. There are at least three factors which are contributing to this qualitative change: the utter consistency and thoroughness of computers, the fact that computers can give us <u>n</u> results for the price of one effort, and the interaction of man and machine in which man thinks while the machine does tedium. In the next three sections each of these points is considered in turn, along with some specfic application areas which exemplify the given principle. Keep in mind, however, all of these principles are involved to some degree in each of the application areas.

3.1 The Utter Consistency and Thoroughness of Computers

Computers, unlike humans, are utterly consistent and thorough in everything they do. They always do precisely what they are directed to do without leaving any instances unprocessed. This characteristic of computers can be used in many ways to improve the quality of linguistic field work.

Consistency checking programs have already been mentioned in Section 2. A program can check literacy primers to ensure that lessons do not contain items which are still untaught, to check the rate at which new words are introduced in lessons, and to check that new items are sufficiently reviewed in following lessons. Programs can check lexical and grammatical files to ensure that entries are properly filled in and that descriptive keywords (such as taxonomic labels) are used consistently. Spelling consistency in vernacular texts, data files, and the linguist's descriptive write-ups can be checked with the aid of computer programs.

The thoroughness of computers is put to good use in programs for generating blank word lists and grammatical paradigms or constructions. Given a correct description of the tactics of the phonological system, or of the formation of a paradigm, the computer will generate every possible form accounted for by the description. It is the thoroughness of the computer that allows us to make orthography changes after texts are already typed and then to have the assurance that every occurrence of the changed item will indeed be automatically changed.

It is the thoroughness and consistency of the computer in grammar testing that will make possible a degree of objectivity previously unkown in grammatical analysis. With computer grammar testing, the linguist knows exactly which utterances are accounted for by the description and which are not. Computer testing also makes it possible for the linguist to know immediately the side effects of any modification to the grammar.

The thoroughness and consistency of computers are opening up new horizons in lexicography. Joseph Grimes (1980) has used a portable microcomputer in Mexico to compile a dictionary of the Huichol language using the meaning text model of language. He is taking advantage of the thoroughness characteristic by having the computer generate all the cross-reference entries implied by each entry in the dictionary. He is taking advantage of the consistency characteristic by using the computer to check that the definitions are not circular. That is, if *a* is defined in terms of *b*, then *b* cannot be defined in terms of *a*.

3.2 *N* Results for the Price of One Effort

The phrase "*n* for the price of one" refers to the principle that given one amount of effort by the human linguist, the computer can take that one input and use it *n* times to produce *n* results. This characteristic of computers promises to make a tremendous impact on the amount a field linguist can accomplish within a given period of time.

A simple example of *n* for the price of one is a lexical card file. When making lexical file slips by manual methods, not only must one make a main entry card for the vernacular word, but one must also make additional cards for each vernacular term that it cross-references and for each gloss that it will be filed under in the inverted gloss file. Thus to get *n* cards, the linguist must exert *n* amounts of effort. (Carbon paper can help, but the results are less than ideal.) However when using a computer, the linguist expends one amount of effort to key in the main entry, then the computer automatically generates the main entry card, the cross-reference cards, and the gloss cards.

Word processing is another application which exemplifies n for the price of one. For instance, when preparing a description for publication, the linguist types the text only once, the first time. After that, the computer does all the typing. The linguist only specifies the changes that need to be made in successive drafts of the paper; that which is alright remains untouched. Thus the linguist expends one amount of energy in typing the complete text once, and the computer multiplies that effort n times as it types out all of the successive drafts.

Matrix permutation is another application here. By this I mean the permutation of the rows and columns in matrix displays of morphological material (such as Pike has worked on) or of statistical material such as showing the degree of similarity within a set of languages or dialects. By manual methods, if one wants to permute a matrix n times, one must spend n amounts of energy copying the whole matrix n times. On the other hand, with the computer one must expend only one amount of energy to key in the whole matrix at the start. After that the computer does all the recopying as the linguist specifies the rows and columns to swap.

The language survey analysis programs shown in Figure 6 are another example of n for the price of one. In a language or dialect survey, one kind of information collected is a list of words in each of the dialects. These comparative word lists then form the basic data for studying the linguistic similarity between dialects. By manual techniques, about all that investigators have been able to do with the data has been to count the percentage of cognate lexical forms. For a large survey, the manual counting of cognates takes weeks, even months. In a computer assisted project, the computerized counting of cognates is only the beginning of the payoff. Once the linguist has spent his time entering the word lists and cognate set decisions, most of the work is done. After that, the computer can use the one basic data file to perform n analyses. After lexicostatistics, the linguist can work on lexical isogloss analysis, phonostatistics, the comparative method, phonological isogloss analysis, and a refined phonostatistics in which the results of the comparative method are used to quantify the degree to which sound change occurs by regular divergence rules and the degree to which it occurs because of social contacts and convergence (Simons 1977a). These analysis techniques have been virtually untouched by our field personnel simply because they are so extremely time consuming by manual techniques. However, with the computer they become almost free once the data file has been prepared for the first step of the analysis.

3.3 The Interaction of Man and Machine

The principle of interaction is based on the idea of a division of labor. There are some things which people can do better than machines, and there are some things which machines can do better than people. For instance, people can think and reason and create better than computers can. On the other hand, computers can do a number of things better than people can, and it is precisely these things which the field linguist finds himself doing most of the time--things such as filing data, alphabetizing files, searching for examples, counting occurrences, typing, formatting data charts, and so on.

The interactive approach to computing divides the labor between the computer and the human, assigning to each what it can best do. The result is that the computer is doing what it does best--tedium--and the linguist is free to do what he does best--think. Basically, the linguist formulates a hypothesis, uses the computer to test the hypothesis, and then evaluates the result. This may lead him to refine the hypothesis, begin again with a new one, or conclude that the result is satisfactory.

Two more terms are useful in describing this approach to programming. First of all, it is a method of partial automation. That is, we recognize that the small computer by itself cannot embody all the experience of the linguist and thus give a final analysis of a large scale linguistic problem. Thus only parts of the analysis are done by the computer, exactly those parts which it can do better than the linguist. Secondly, it is a method of successive refinements. That is, we recognize that the solution cannot in general be found the first time the data are processed. Thus the results of the first pass are used to sharpen the hypothesis so that a second pass can be performed, and so on. The analysis is successively refined at each step.

A generalized scheme for the interactive approach with partial automation and successive refinements is as follows:

(1) The linguist decides what he needs to study and selects the data corpus.
(2) The computer finds, arranges, or computes some intermediate results from the data corpus.
(3) The linguist inspects the results and makes a hypothesis.
(4) The computer runs a program which tests the hypothesis.
(5) The linguist decides whether the data fit the hypothesis. If they do not, either the linguist refines his hypothesis and goes back to step 4, or he goes after more data by returning to step 2.

Now I shall describe how this process works for four specific application areas: word processing, primer sequencing, comparative method, and grammar testing.

Word Processing

(1) The linguist types in the text. Go to step 3.
(2) The linguist enters changes into text with an interactive editing program.
(3) The computer prints the complete text with automatic formatting.
(4) The linguist makes a decision: if the printout is adequate he publishes it, otherwise he reads and marks the printout and returns to step 2.

Primer Sequencing

(1) The linguist with the aid of the computer has already come up with productivity counts on the orthographic symbols, and syllable shapes, and counts and analysis of the grammatical functors.
(2) The linguist posits a primer sequence specifying the order in which the symbols, syllable shapes, and functors will be introduced in the primer.

(3) The computer tests the sequence against the lexicon, or against a special word list derived from a corpus of texts, and lists all the words available for examples at each step in the sequence. If functors are included, the computer can search texts to find complete phrases and sentences available at each step.
(4) The linguist makes a decision: if the forms available at each step are sufficient to make good lessons, then work on the primers begins; otherwise, the linguist returns to step 2 to propose another sequence.

Comparative Method

(1) The linguist, with the help of an interactive program, formats the comparative word lists for phoneme by phoneme comparison.
(2) The linguist selects a minimum correspondence value for this pass through the analysis. Initially it is 0% and raised through successive passes as non-related forms are weeded out. (Each set of possibly cognate words has a correspondence value assigned to it. It tells the percentage of phoneme correspondences in the words which are regular. 0% means not at all regularly related; 100% means the words are completely regularly related.)
(3) The computer tabulates frequency counts for all the phoneme correspondence sets occurring in cognate sets with at least the minimum correspondence value. (This requirement serves to weed out non-cognate sets of words from the corpus.)
(4) The linguist inspects the results (calling on the computer to find examples of the correspondence sets when needed) to determine correspondences that can be established as regular.
(5) The linguist makes a decision: if no new correspondences were established and the analysis is finished, go to step 7; if no new correspondences were established but the analysis is not complete, return to step 2; otherwise, continue the analysis with step 6.
(6) The computer assigns new correspondence values to all the sets of possibly cognate words on the basis of the new list of established correspondences. Return to step 2.
(7) The computer lists all the established correspondences with examples and all cognate sets with their final correspondence value.
(8) The linguist posits the regular sound correspondences and the sets of words which are indeed cognates.

Grammar Testing

(1) The linguist has already prepared a formal grammar description, a lexicon, and sample forms (phrases or clauses or sentences and so on), and entered them into the computer.
(2) The computer parses the sample forms using the grammar and lexicon specified by the linguist. The result is a structural description of the forms.
(3) The linguist makes a decision: if the structural descriptions are correct, then he is done (or he gets new forms and returns to step 2;) otherwise, the linguist refines the formal grammar or the lexicon and repeats step 2.

4. SUMMARY

The three sections of this paper can be summarized as follows: (1) The microelectronic revolution is an incredible technological breakthrough which has made on-site computing for the field linguist possible. (2) A computer can assist the field linguist in almost every phase of his daily work, from the beginning of a project right up to its completion. (3) On-site computing is bringing about a qualitative change in field linguistics, because of the utter consistency and thoroughness of computers, because they give us n results for the price of one amount of effort, and because the interaction of man and machine is making possible a division of labor in which the linguist thinks while the computer takes care of all the tedium in analysis work.

BIBLIOGRAPHY

Grimes, Joseph E. 1967. Positional analysis. Language 43:437-44.

Grimes, Joseph E. 1974. Dialects as optimal communication networks. Language 50:260-9.

Grimes, Joseph E., ed. 1975. Network grammars. Summer Institute of Linguistics Publications in Linguistics, and Related Fields No. 45.

Grimes, Joseph E. 1980. Systemic analysis of meaning. Notes on Linguistics

Grimes, Joseph E., John R.Alsop, and Alan C. Wares, 1968. Computer backup for field work in phonology. Chicago, University.

Grimes Joseph E., Ivan Lowe and Robert A. Dooley. 1978. Closed systems with complex restrictions. Anthropological Linguistics 10:167-83.

Henderson, J. E. 1975. Programmed review cards. Language Learner's Field Guide, edited by Alan Healey. Summer Institute of Linguistics: Ukarumpa, Papua New Guinea. Pp 318-321.

Longacre, Robert E. 1864. Grammar discovery procedures. The Hague: Mouton

Murdock, George Peter. 1950. Outline of cultural materials, 4th ed. New Haven, Conn. Human Relations Area Files, 1950.

Noyce, Robert N. 1977. Microelectronics. Scientific American 237(3):63-69.

Simons, Gary F. 1977a. A package of computer programs for the analysis of language survey word lists. Working Papers for the Language Variation and Limits to Communication Project, Number 7. Ithaca, NY: Department of Modern Languages and Linguistics, Cornell University.

Simons, Gary F. 1977b. A user's manual for PTP--the Programmable Text Processor. Working Papers for the Language Variation and Limits to Communication Project, Number 1.

Simons, Gary F. 1979. Language variations and limits to communication. Technical Report Number 3. Department of Modern Languages and Linguistics, Cornell University.

Weber, David and William Mann. 1979. Prospects for computer- assisted dialect adaption. Notes on Linguistics Special Issue Number 1.

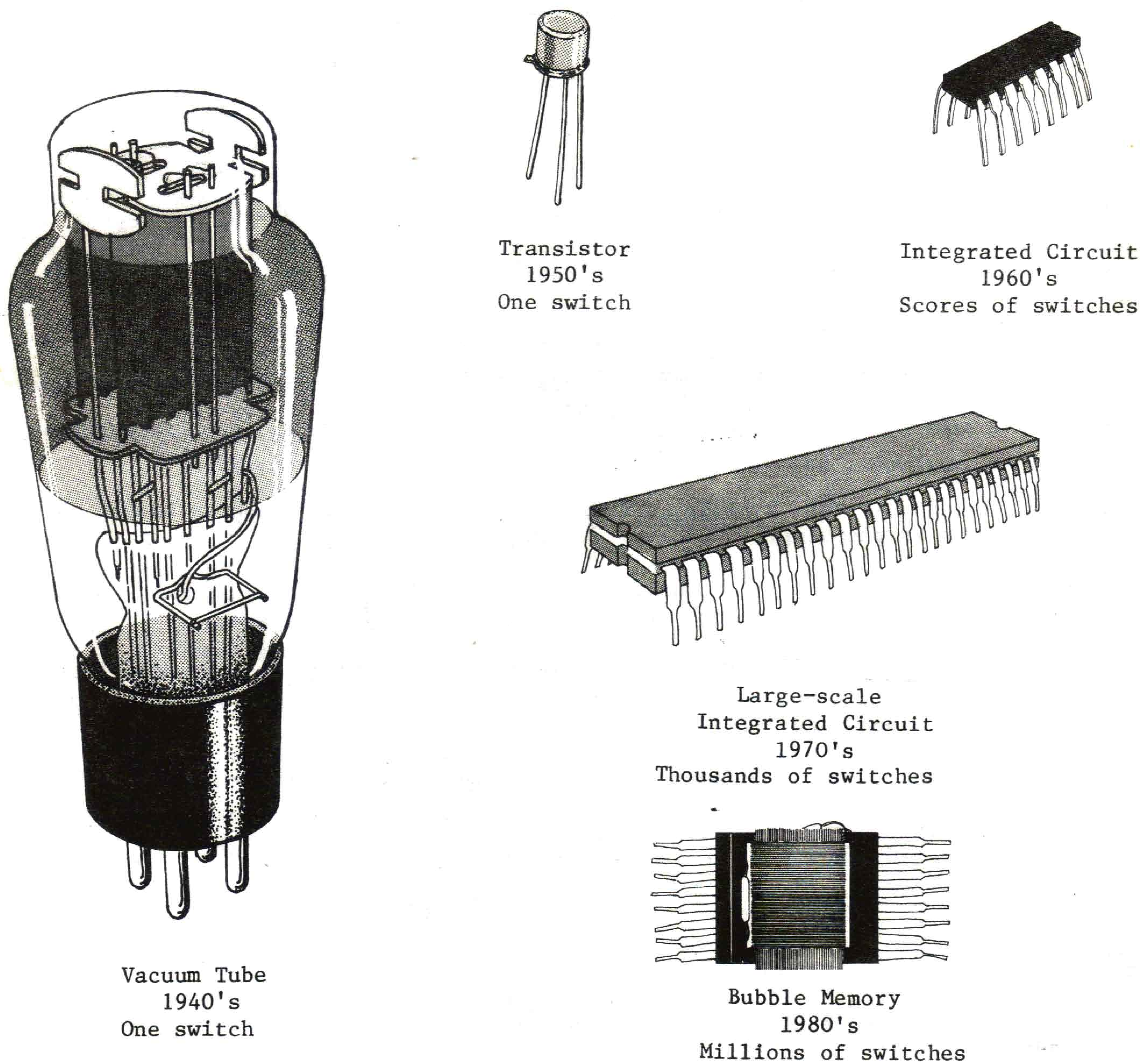

Figure 1. Historical development of electronic components
(Drawings are actual size)

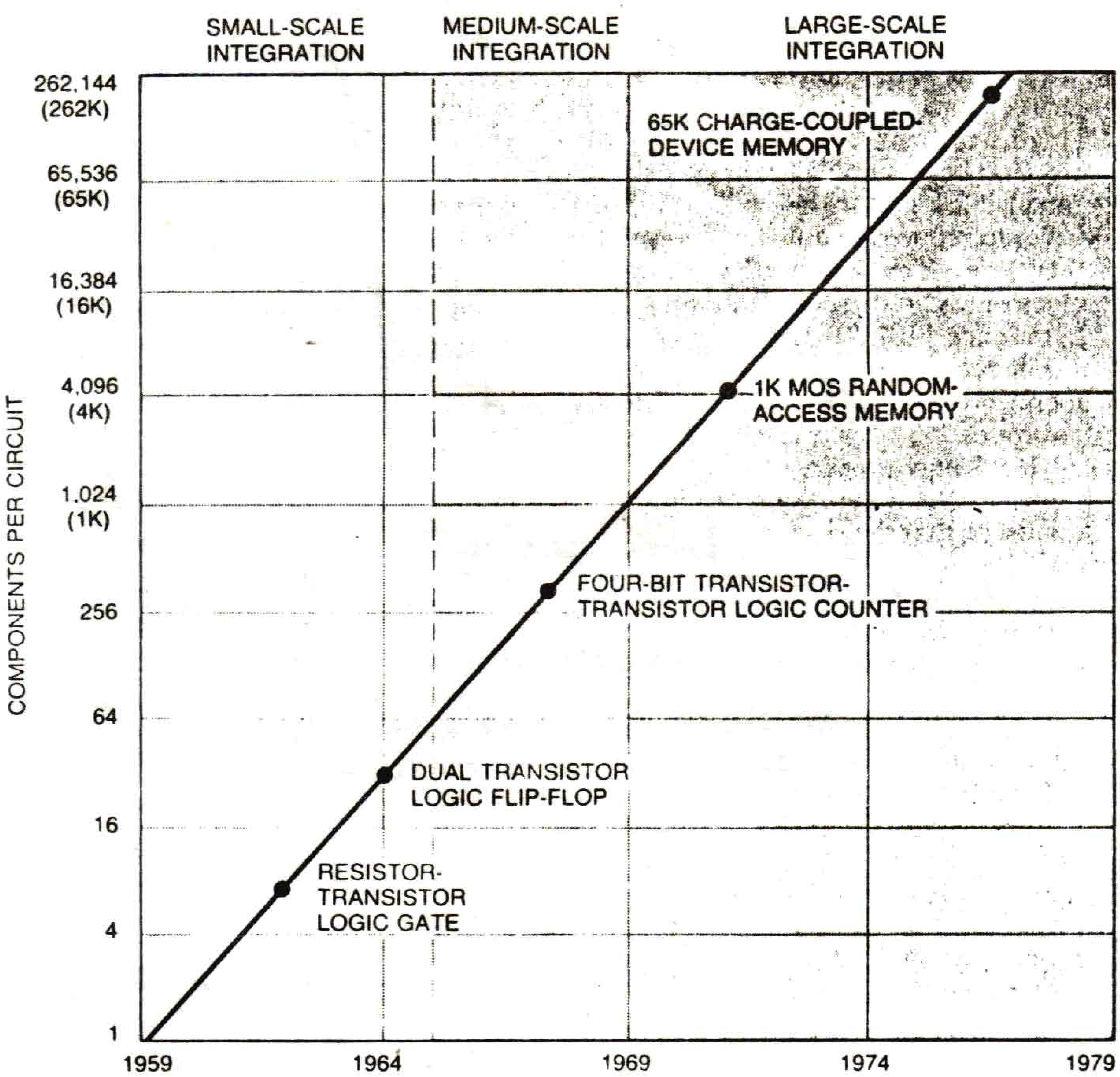

Figure 2 - Number of components per circuit.

From 'Microelectronics', by Robert N. Noyce.

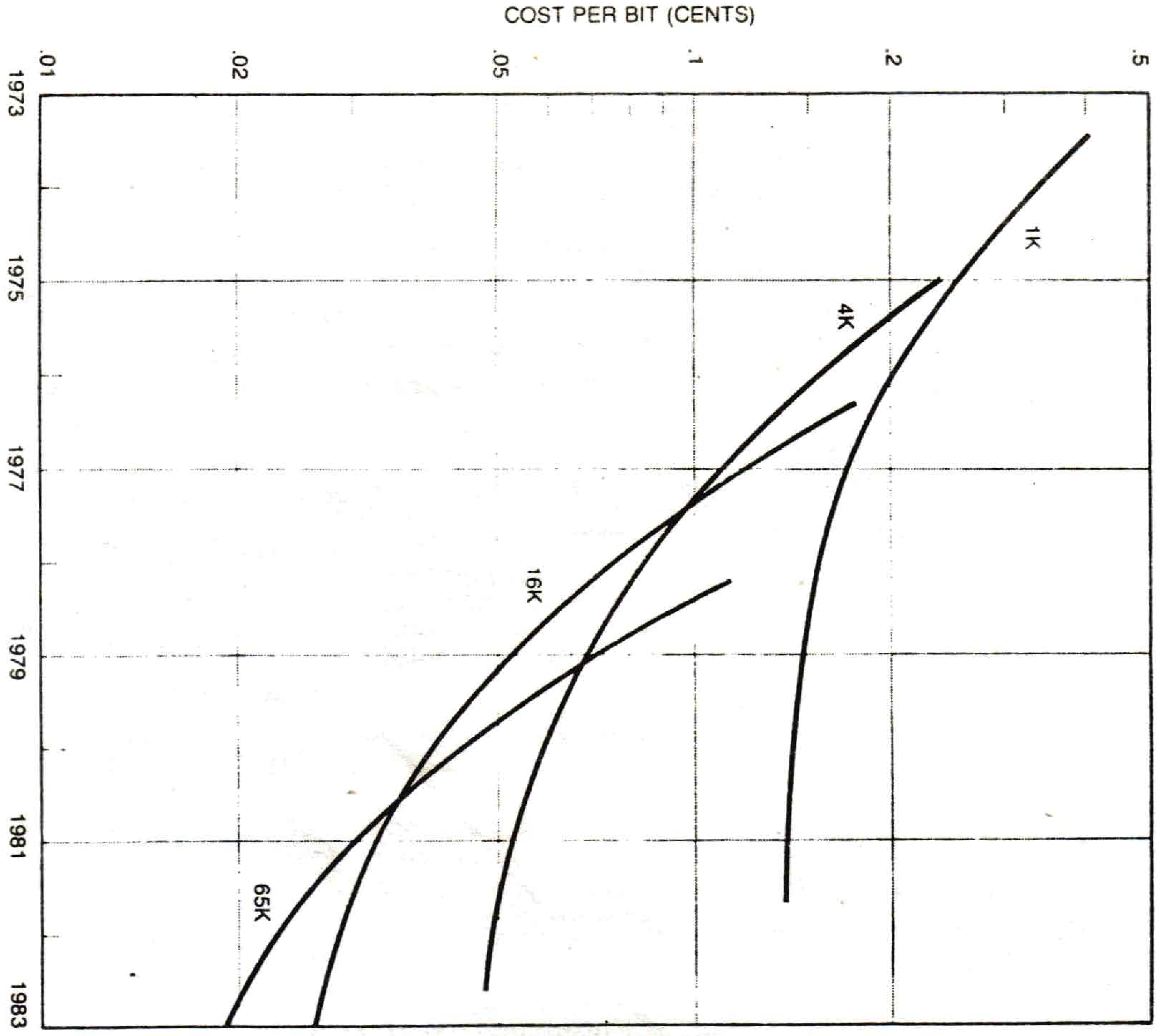

Figure 3 - Cost per bit of computer memory.

From 'Microelectronics', by Robert N. Noyce.

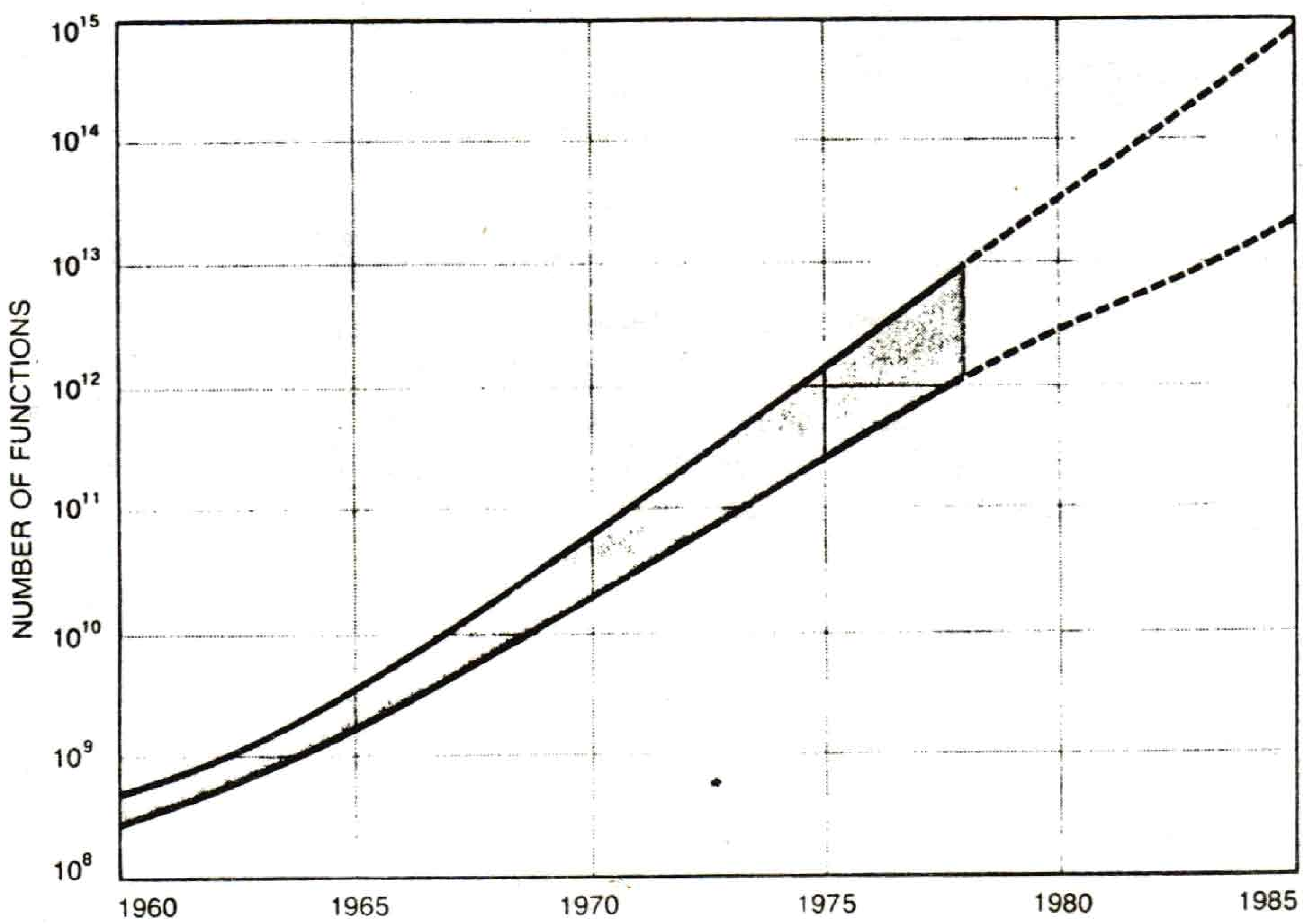

Figure 4 - Annual utilization of electronic functions.

From 'Microelectronics', by Robert N. Noyce

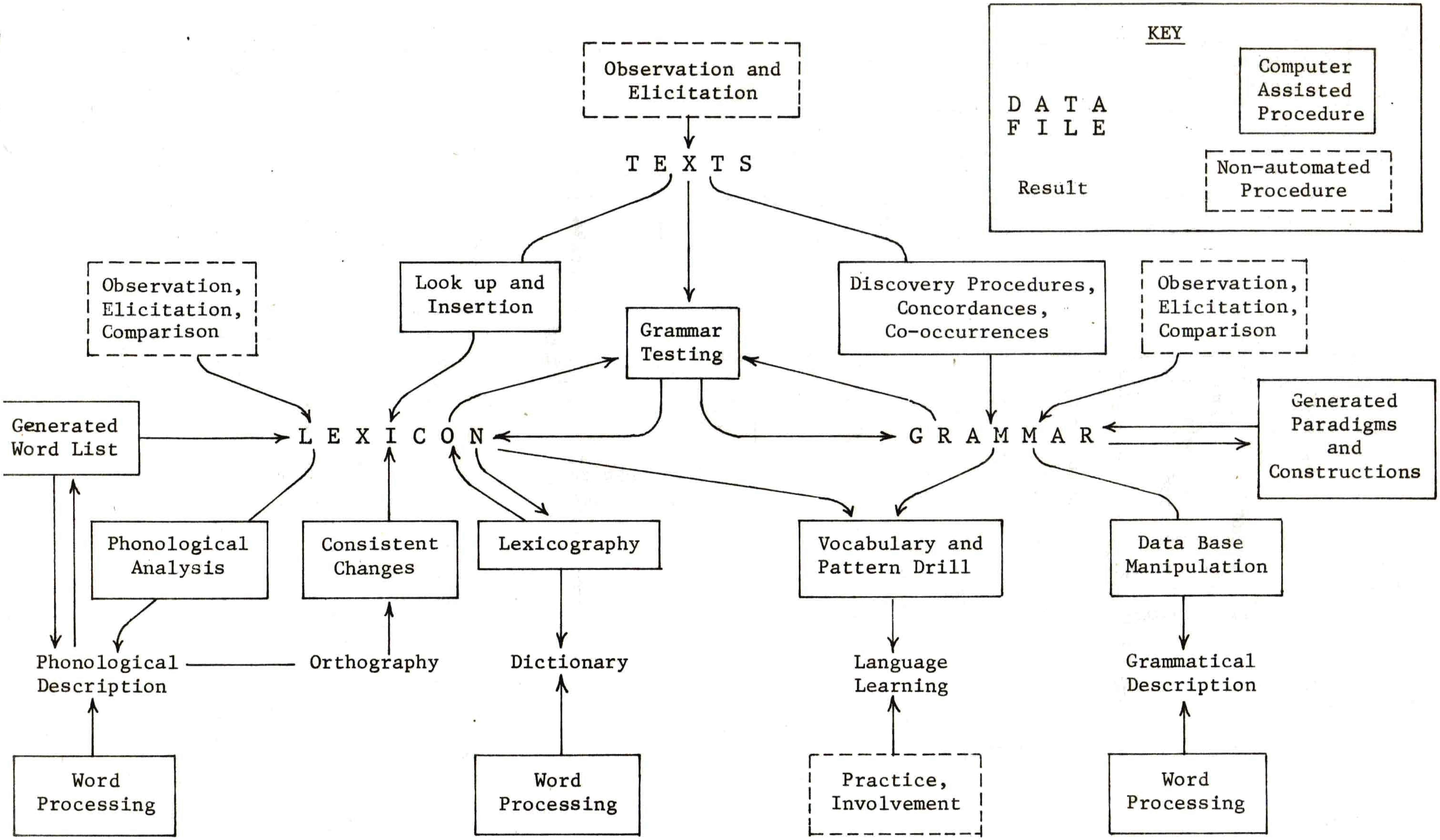

Figure 5. An Overview of Computers in Field Linguistics.

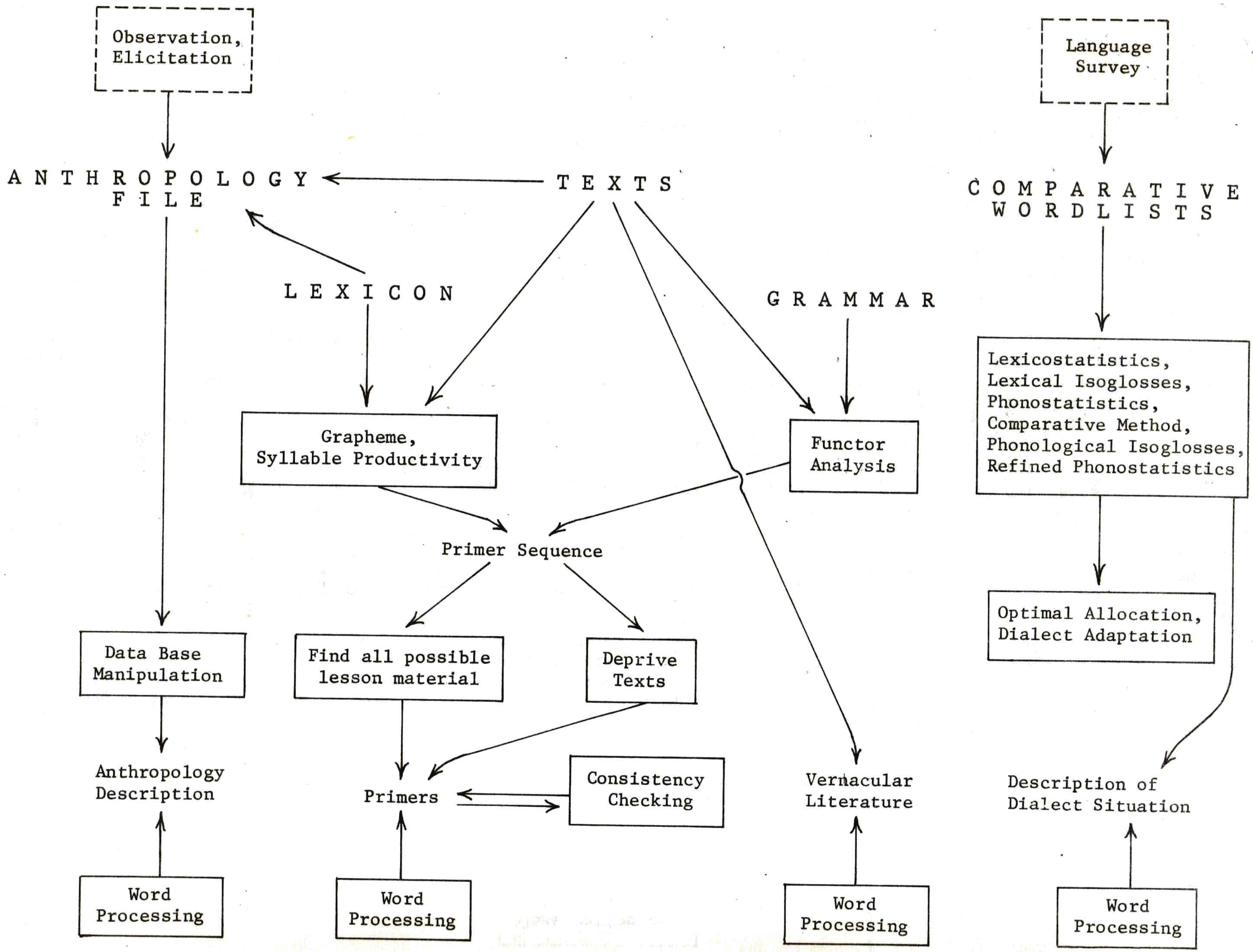

Figure 6. An Overview of Computers in Anthropology, Literacy, and Language Survey.