AREA Conference
February 1965

Remarks by Sally Y. Sedelow
Saint Louis University

Learning the New Methodologies

Professor Page has charged me with the task of talking both about the announced topic in the program, Learning the New Methodologies, and about my own research.[1] In order to make for a meaningful connection between the two, I shall first provide a very broad and brief description of the stylistics research project and then speak about acquiring the techniques or methodologies necessary to pursue or make use of such research.

The stylistics project is directed toward the development of a programming system which will produce a multi-dimensional analysis of the style of any piece of writing, whether prose or poetry, whether written for example, by an elementary school student, a neurotic or psychotic, a college student or a professional in any field. By style we (my husband, Walter, and I - and Terry Ruggles, a programmer) mean the "patterns formed in the linguistic encoding of information." Such patterns include word choice as well as arrangement; therefore, we are interested in what is sometimes called content analysis as well as in other aspects of style. In fact, of the two programs we have designed and are currently using, one of them VIA (verbally-indexed associations) is especially designed for analysis of word choice; the second program, MAPTEXT, provides abstract representations of any specified textual element (or elements) as well as statistical summaries.

The difference between VIA, our content analysis program, and the General Inquirer, for example, reflects the differing goals of the research

[1] This research is supported by the Office of Naval Research, the System Development Corporation, and Saint Louis University.

for which the two programs are conventionally used. Because of the precompilation of the dictionary and tagging of words as to concept association, The General Inquirer is especially well suited for the testing of a particular theory or for measuring verbal data against the verbal components comprising a particular theory -- for example, a psychologist reviewing patient protocols will adduce the presence and association of certain concepts as evidence of a specific type of aberrant behavior. Like the psychologist, The General Inquirer can be pre-set to tag certain words as indicative of fear, hate, etc. Because we are interested in determining what is distinctive about any given piece of writing, VIA is set up to work on a reflective, responsive, adaptive basis. VIA, too, constructs lists of words which are conceptually associated but VIA's thesaurus grows from and is based upon the individual text being examined. VIA, in other words, is meant to simulate the so-called objective critic -- one who is not looking for something in particular when he begins reading the paper or document or whatever, and who adjusts his categories dynamically as a function of feedback from the text, itself. So far, VIA has been tested on two very different types of text: _Hamlet_ and two translations of a compilation of articles, by Russians, on Soviet Military strategy. Currently, an extensive and intensive study is under way with reference to Hume's _History of England_.

As an indicator of major themes or concepts, VIA has done very well on both Soviet military strategy and _Hamlet_. Having conducted extensive manual, or traditional, investigations of themes in _Hamlet_ as well as having carefully examined similar studies by other critics, I am convinced that VIA turns up most, if not all, of the themes which have been thought important as well as others which no one seems to have noticed but which would, nonetheless, considerably enhance interpretations and studies of the play. (For example, there is a pattern of familial designations in the first four acts of the play which significantly vanishes in the last act.) Even with current parameters (for an explanation of this term, see the Annual Report mentioned below), VIA's output will result in more complex, subtle analyses of themes in Hamlet than have hitherto been available. A change in the parameters would produce many additional linked words and lists and thus the opportunity for an even more detailed examination of the interplay and interrelationship among themes.

VIA also reveals other aspects of style. For example, VIA clearly shows whether differences of word choice between the two translations of Soviet Military Strategy are matters of concept or word preference: for instance, the research scholar might think at first glance that the absence of the word "conflict" from the first chapter in one translation and its use twenty-four times in the other translation could be indicative of a real conceptual disparity between the two translations; but then he would notice that the translation lacking "conflict" includes, in a similar conceptual word list, the word "combat." A further glance at the relative frequencies of the word "combat" in the two translations shows that "combat" occurs twenty-odd times more often in the translation laking "conflict." The two translations, then, are almost equally militant; there is no conceptual disparity so far as these words are concerned. There are many ramifications to VIA's output, but there is not time to mention them. A detailed explanation will be provided in the Second Annual Report on the Stylistic Analysis Research Project, to be available about March 1.

If the research scholar chooses, VIA's output can be used as the basis for input into our second program, MAPTEXT. For example, we wanted to study tbe distribution patterns of the word "general" and its synonyms. MAPTEXT produces a graph showing the distribution as well as statistical summaries of the distribution -- by comparing the graph of "general" and its synonyms, we can quickly see whether all the words are similarly distributed (thus implying some sort of conceptual grouping) or whether, when the use of one word declines the use of a synonym increases.

I should think that programs such as VIA and MAPTEXT would have considerable utility for all kinds of educational purposes. Perhaps, for example, some version of VIA combined with Ellis Page's measures for essay grading can provide a teacher with fairly comprehensive evaluations of the quality, both as to content and other variables of style, of student themes, MAPTEXT can be used to provide visual guides to shifts in all kinds of usage patterns -- ranging from word choice to punctuation choice. MAPTEXT could provide quick aid to the teacher interested in spot checks on particular trouble spots in an individual student's writing((for example, does the student's word choice lack variety, do his sentences too often begin with a personal pronoun, etc.). MAPTEXT could also be used as another sort of pedagogical aid by providing the student with precise graphical displays of a given author's characteristic style or of attributes of that style. For example, a version of MAPTEXT has been used by graduate students in English at UCLA to study patterns of the caesura in the writings of given authors; and a program analogous to MAPTEXT is being used by Prof. Jan LaRue, of New York University, to study patterns of musical theme -- he has suggested that this program be called MAPSCORE.

Now, as to acquiring the methodologies so that you can use these and similar procedures for educational research and, ultimately, for education -- the first question is: should you know how to program the computer? For the foreseeable future, the answer is yes, you should know something about programming computers. At present, except for standard statistical procedures, programs of use for education and educational research are in their infancy but within a decade they can be expected to move through adolescence into fully-developed maturity. In order to evaluate the suitability of existing programs for one's own needs, it is necessary to have some notion of the program's operation; otherwise, one is likely to be either confused or overwhelmed by performance claims and thus expect results which the program may not provide. Having decided that a particular program looks promising, the prospective user then may discover that it was written for a computer his own institution does not have and in a language no home-institution computer speaks. Again, it is useful to be able to make some personal evaluation as to the feasibility of putting the program into a form comprehensible to the computer to which one has access. Computer center people will be glad to help but they, as a rule, are not fully versed in problems attendant upon verbal data processing and, therefore, it is well to be able to weigh their advice. Most important of all, the existing program--and since they are in the main experimental, this will doubtless be true of programs for the foreseeable future--probably does not provide exactly the output you want. Again, some knowledge of programming will enable you either to provide personally for the output or tell someone else i.e. the programmer, how to do it. In many cases, there may well be no program that suits your needs--or you may be conducting research that requires continuous development and change in the programming system. In these cases, it is especially important to know something about programming so that you have a clear sense of the computer's capacities as well as some command over them. I don't suggest for a moment that you will want to do all or maybe even any of your own programming--but you *will* want to know, in some detail, just what your programmer is doing--otherwise you may discover that, in order to suit the computer's convenience, an important part of your research procedure has been distorted-- or, even worse, you may *not* discover that part of your research procedure has been distorted. So, for all these reasons, I believe a basic understanding of programming is extremely desirable.

Having established this desideratum, the next question--at least from students in literature--usually is, must I,as a prerequisite, know mathematics? The answer is, no, you do not need to know mathematics in order to learn to *program* a computer. Unfortunately, most computer programming courses and manuals seem designed on the premise that one *does* know mathematics; the examples and explanatory text are often very strongly mathematically oriented. The reasons for this slant initially are obvious--computers were used for mathematical manipulations. Now, I suspect, something of a professional clique has developed, with a notion that, "Now that we have a good thing, let's not let the field become too crowded. One obvious way to reduce the number of entries is to set up course prerequisites that many people cannot or will not meet; and, if they should try to teach themselves, there's the added obstacle of relatively incomprehensible manuals and books.

Having initially gone through a programming course designed for people who intended to use the computer to track airplanes, I am trying to work out procedures which will make programming easier to learn and, concomitantly, more interesting for people who are especially anxious to use the computer for the analysis of verbal data. At Saint Louis University, I teach a graduate course for students in the humanities and social sciences who want to learn to program the domputer. The course is divided into two semesters. During the first semester, the students learn basic programming procedures and specific programming languages; the second semester, which is optional, is a seminar-workshop. Students who intend to use the computer for doctoral dissertations or master's theses may work out the programming procedures and do the coding during this time; other students may choose to research a particular area (such as parsing systems, or content analysis) and design small experimental programs pertinent to that area. As an example of the former, one student is using the computer as an aid toward graphemic analysis of some Middle English manuscripts. As an example of the latter, another student tested out some ideas about grammatical transformations by simulating them on the computer.

To be more specific about the teaching of actual programming techniques: we begin by learning how to sort verbal data (to alphabetize it, for example), transfer it, and search it through general flow-charting techniques. Having mastered the general procedures, the students than translate them into specific languages. I try to teach a procedure-oriented language (FORTRAN, currently), a machine oriented language (SPS) and, ideally, a list-processing language (SLIP). The examples, of course, are always verbal. The students alphabetize, search for function words, make indexes, and, the more advanced may write a concordancing program. In addition to their work with programming languages, they research relevant literature and make reports on their findings (the latter activity is necessary to keep broader goals in view and thus provide a stimulating context for the highly specific and detailed programming activity). This is all done in a standard 3- credit hour course in <u>one</u> semester. Although the course is currently a graduate course, the first semester could profitably be offered to undergraduates as well -- even, I think, to high school students. And if some experiences with the teaching of the "new mathematics" signify what I think they do, it may be that we should -- as with second language learning -- begin with the very young indeed for the best and most exciting results. I have descriptions of the course which are a trifle more detailed tha that I've provided during this short talk. I might also note that my husband and I are producing a textbook on verbal-data processing, to be published by McGraw-Hill.

It's my conviction that before too many years have passed almost every scholar in every field will need to know something about computers. In some cases, for those who fear and despise the "machine" this knowledge will be a matter of self-defense. But in many cases, the scholar will have learned that the computer is a major research tool and clerical aid for the processing and analysis of verbal data. One major advance will be the shift from inhumane use of many teachers and graduate students (many educators who hate machines don't object to using humans further down the organizational ladder than themselves as machines) to the much more appropraite non-human machine. But the more major gain will be the use of the computer to provide new insights into the organization and nature of knowledge, itself.