DISCUSSION

Tore Dalenius, Brown University

Had Charlie Chaplin observed the processing of a census, his classic film "Modern Times" might have been shot at a statistical agency rather than a manufacturing plant.

The U.S. Bureau of the Census pioneered the use of computers in surveys/censuses. It suffices to mention applications to editing of data and production of statistical tables which originated in the 1950s. This meeting focuses on a new area, viz. coding for survey/census responses -- again, the U.S. Bureau of the Census has been a pioneer.

Automated coding has already successfully passed some practical tests. It is, however, not a complete methodology; further research and development is called for. The authors have provided a comprehensive account of some of the problems and ways of addressing them which should prove most valuable in future work, say for application in the 1990 censuses of population.

The three papers have much in common with respect to problems and methods, and reflect, I surmise, several years of close cooperation between the two agencies involved. Consequently, I have chosen a rather general format for my discussion.

1. The Coding Operation

Coding calls for assigning each element in a survey/census to one of several mutually exclusive classes, with associated codes and code descriptions.

The data available for this operation is the natural language responses (NLR) to one or more questions. The NLR must be processed in terms of their subject-matter meaning. This calls for generating linguistic constructs (LC) which may be used to link an element with a code.

Traditionally, this linking has been made by clerks. Experience shows that clerical coding has several important characteristics. Especially, on the positive side, it allows coding of "difficult" cases. On the negative side, it is expensive, time-consuming, error-prone, difficult to manage and control, and boring -- characteristics which make it natural to look for an alternative approach, viz. automated coding.

The starting-point of all three papers is the same kind of NLR as available for the traditional clerical coding. They then go on to discuss, in considerable detail, the generation of the LC which serve as the variables of the coding algorithms.

The automated coding is carried out in three steps:

i. reading in each NLR;
ii. generating the corresponding LC; and
iii. assigning the unit of analysis to one of the codes.

Of these steps the generation of the LC is the fundamental one.

It is clearly highly desirable, from an efficiency standpoint, that this generation be carried out by the computer. The authors discuss a variety of schemes (algorithms) which have been developed in the course of their work during more than a decade. It is common to these schemes that they call for access to a "knowledge base", which may be a sample of clerically coded NLR, perhaps supplemented by other material.

Automated coding would be no difficult task if, for a given NLR, there is an easily identified linguistic construct LC which (according to the knowledge base) is associated with a unique code C. In many instances, however, the situation is different. We will give two examples. Thus, a NLR may yield two or more LC. Or a LC may be associated with two or more C.

I will now consider the performance of the algorithms discussed. In doing so, I will not enter upon a comparison of the performance of the U.S. Bureau of the Census's algorithms and Statistics Sweden's algorithms, as the operational settings are so different. The authors pay special attention to the coding rates and the error rate (relative to clerical coding).

The discussion in the papers suggests that with a coding rate of about 75%, the accuracy arrived at is acceptable, and that this coding rate is sufficiently high to justify automated coding. Algorithms which yield a significantly higher coding rate (say above 90%) tend to yield an error rate that is by far too high to be acceptable. There is clearly an unfavorable relation between the coding rate and the error rate. The situation appears to be different from the case of automated editing. There it has been found that it is possible to keep the error rate at a low level while achieving a high editing rate. In fact, automated editing has proved highly efficient, to the extent that it has been possible to use it as a replacement for clerical editing rather than a supplement.

The fact that a 75% coding rate is sufficient today reflects, of course, the relative cost of human labor and use of the computer. The relative cost may be less favorable in the near future: the cost of labor may increase and the cost of using the computer may decrease, making it desirable to increase the coding rate.

In what follows, I will identify three lines for future work. I will also point to a non-technical aspect of which the authors are no doubt already aware.

2. Three Lines of Future Work

I will present these lines in decreasing order of expected potential usefulness.

Line No. 1. I am convinced that the methodological situation is already sufficiently satisfactory when it comes to getting access to powerful algorithms, given highly informative LC (in the sense implied by the papers). I am, however, equally convinced that the prime emphasis of future work should be on the linguistic problem. I propose that the emerging discipline called "computational linguistics" will in the near future advance to a stage which makes it practically useful in the context of automated coding. Reference (1) may prove of interest. More specifically, research concerning the construction of a "reverse dictionary" presently being carried out by Professor Henry Kucera at Brown University

may prove applicable to automated coding, as the following simple illustration shows (expressed in terms of automated coding): Given the NLR = "fear of heights", the dictionary provides the LC = "acrophobia" -- which operation is, it seems to me, close to automated coding.

A closely related question is as follows: Should the data collection in a survey (census) be designed to reflect the problems of generating linguistic constructs? It is my opinion (but perhaps I am wrong) that the answer is Yes. In this context, it may be useful to consult the literature on the pros and cons of "open-ended questions."

Line No. 2. What is the potential of parallel processing? It is a striking observation that while a computer carries out individual operations faster than the human brain, the latter nonetheless may solve a complex problem (much) faster; this is due to the fact that the human memory is associative. As an example, if I say "Charlie Chaplin," you may associate this name with his shoes, or his stick, or his moustache, or one of his films. Parallel processing aims at mimicking the human brain in its operation. There is reason to believe that in the not-too-distant future, schemes will be developed which permit the computers to make inferences: to complete state vectors (which would be of interest for editing/imputation), and to infer from one state vector to another (which would be of interest for automated coding).

Line No. 3. It may finally be worthwhile to look for applications similar to automated coding in other non-statistical fields. One possible field is automated medical diagnosis. References (2) and (3) may provide ideas.

I hasten to add that medical diagnosis typically is carried out in an interactive fashion: the physician observes some symptoms and proceeds accordingly. This fact may serve to make these applications less interesting in the context of automated coding.

3. A Non-Technical Aspect

Finally, I will return to Charlie Chaplin. In recent years, concern has been expressed about the impact of automation on the quality of life. It may be worthwhile to pay attention to this aspect when you continue your endeavors in the realm of automated coding.

References:

(1) Berwick, R.C. Computational complexity and lexical-functional grammar. American Journal of Computational Linguistics, vol. 8, no. 3-4, July-December 1982, 97-108.

(2) Kulikowski, C.A. Artificial intelligence methods and systems for medical consultation. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-2, no. 5, Sept. 1980, 464-476.

(3) Szolovits, P. and Pauker, S.G.: Categorical and probabilistic reasoning in medical diagnosis. Artificial Intelligence, 1978, 115-144.