Papers Presented to 7th International History Congress, August 1978

Computer input of Historical Records for Multi-Source Record Linkage

C.J. Jardine and A.D.J. Macfarlane, Cambridge, UK.

Introduction

This paper describes a method for the preparation of historical records for computer analysis. The method has been developed as part of a larger study of social and economic structure in the past.

The official records which have come down to us from the sixteenth, seventeenth and eighteenth centuries contain a wealth of detail about the social and economic circumstances of ordinary people, but this material is hard to use. The information was originally collected and recorded for a variety of different purposes (criminal law enforcement, taxation, the registration of landholdings etc.). Each source gives a series of isolated details which must be reorganized to form a coherent whole if their full potential is to be exploited.

The work of Fleury and Henry and their followers has demonstrated that the information contained in a parish register becomes much more useful if it is reorganized, by the method of "Family Reconstitution", so as to provide family biographies (See bibliography in Wrigley ed. (1973)). Whereas the data in their original form could only provide estimates of crude vital rates, the reorganized data yield age specific rates, birth intervals and the rest. The essence of the method of Family Reconstitution is the bringing together of relevant information from different sources, or from different parts of the same source. If a baptism record and a marriage record for the same person are brought together, an estimate of an age at marriage results. An item of information has been created by the juxtaposition.

The project of which this paper describes a part aims to use the collating power of computers to discover as much as possible about early modern social and economic structure. The method is similar to family reconstitution in that the data are reorganized principally by person, but differs in that all available sources are used, rather than only parish registers. Most of the sources contain a great many names - the accused and witness in court cases, previous and present tenants in a rental, testator and beneficiaries in a will. By bringing together references to the same person from different sources one may collect biographies of considerable length and detail. Many correlations emerge which were invisible in the original sources.

It must be emphasized that the use of a computer is not logically essential for this task. The entire process of collation can be carried out by hand, and indeed has been. We have indexed the material for Earls Colne in Essex over the period 1550-1750 by person and by plot of land independently of the computer. The result is several card index files containing between them well over a million cards. These manual indexes, and the processes involved in their creation are described in Macfarlane, Harrison and Jardine (1977). The central file in this system contains all personal references derived from the sources sorted by surname, forename, and date of birth. Not only was this file very laborious to create, is is also extremely laborious to use. For many purposes reference must be made to separately stored full transcripts of the sources. For many potential uses of the data, the help of the computer is a practical necessity forced on us by the very bulk of the data.

In order to use a computer to process historical data we have, as a first stage, to prepare the data for computer input. This means that the data must not only be transformed into a machine readable form by some form of key punching operation, but also that they must be rendered machine comprehensible. The data must he presented to the machine in a physical and logically acceptable

form. It is this matter of logical acceptability which is the subject matter of the remainder of this paper.

Input formats and data models, a digression

Computers have not been with us long: about twenty-five years. During this quarter century enormous advances have been made in electronic technology. These advances have had the effect of dramatically reducing the physical size, and the cost, of computing machinery while increasing its power and complexity. In response to the rapidly increasing availability of computing power there is a slower development of more sophisticated ways of using these enormous machines.

The traditional way of using a computer is problem oriented. The computer is programmed to perform a specific task, and given exactly the data required for it performance. Now that computers are being used for more and more such mundane tasks many institutions find that there is considerable duplication of information within their various computers. Different parts of the organization may use the same data for different purposes. The separate computer programs serving these different purposes, while they may require the same data, say names and addresses, will require them in a different form and different order. This situation can result in considerable expensive duplication of effort as data are collated and prepared more than once.

Realization that this is happening has lead to a change in attitude whereby data themselves are regarded as a valuable commodity. Ways are being found of using the computers themselves to transform data from the form required by one program to the form required by another. Attempts are being made to produce suites of computer programs which are capable of performing all conceivable transformations on a body of data. Such a suite of programs is called a Generalized Data Base Management System. For a survey of the literature on this subject the reader is referred to Date (1975).

Investigation in this field has highlighted the importance of the concept of a data model. To take a concrete example which may be familiar to some of the readers of this paper let us consider the Statistical Package for the Social Sciences (SPSS). This is a suite or package of computer programs which are capable of carrying out most of the statistical computations social scientists require. All the separate programs in the suite accept data in the same form. The data model used by the package is as follows. The population under analysis consists of a number of cases, each of which is described by the values of a number of variables. Variables may either have numeric values or alphabetic values. This model is flexible in two important ways. First, data prepared according to the model can be used in conjunction with any of the large and growing number of programs in the package. Secondly, a wide variety of different sorts of data can be fitted into the model. The entity in the real world which corresponds to a case in the model can be almost anything. The data model of SPSS provides a convenient conceptual fixed point, through which a variety of programs can operate on a variety of data. The power of the model is increased by certain programs in the package which transform data. The operations available include definition of new variables in terms of old ones, recoding of existing variables, selection of subpopulations of cases by criteria defined in terms of the values of the variables, and sorting of the population in an order based on the values of variables. These data transforming operations add to the flexibility of the package, allowing the statistical programs to he applied to data indirectly derived from the input data.

This flexibility is, however, limited. An example may make this clear. Suppose that data are available from a cross sectional analysis of children in families. Data about the children include ages and educational achievements; data about the families include their incomes and social classes. With these data it should be possible to investigate such correlations as that between income and number of children for families or that between educational achievement and birth order for

children. For the first correlation we need to consider a population whose cases are families, for the second the cases must be the children. SPSS provided no means of transformation between these views of the data.

These shortcomings of the data model may be of considerable practical importance. In a case like that considered above the investigator is presented with three alternatives; not to use SPSS at all, to write his own programs to perform the required transformations, or to duplicate the effort of data preparation, punching once with families as cases and once with children as cases.

We have a large body of data to prepare for machine processing, and comparatively little advance knowledge of the types of analysis we shall require. We are therefore anxious to avoid duplication of data preparation.

In this discussion of data models nothing has so far been said about the way in which data are to be presented to the machine. The SPSS package provides two formats in which the data may be punched. This choice of input formats illustrates independence of the concepts of input format and data model. Even when an adequate way has been found for modeling data for analysis within the computer, it is a separate problem to design a convenient form for preparation of the data.

These remarks about SPSS are intended only to illustrate some of the consequences of choice of data model in the context of programs which may be familiar to some readers. It is clear that the package is not, and was never intended to be, suitable for the type of analysis we have in mind.

When one comes to look for existing data models, packages or input systems which are suitable for our sort of data one finds that they are very uncommon indeed. The literature on generalized data base management systems, contains much theoretical discussion about data models, transformations of data etc., but very little discussion of ways in which data in large quantities should be prepared for loading into data bases. In commercial practice the computer system designer has considerable control over the process of data generation. Historians have to deal with data coming down from the past. If the data were not collected in a way which suits the computer program, the program must be changed. We cannot go back in time and collect the data in a different way. This is a special problem, and requires unusually flexible computer systems to exploit what data there is. The paymasters of the computer world are large commercial organizations which do not share these problems, so comparatively little work has been done on their solution.

All existing systems for the input of historical data use data models which involve the advance specification of a template. This template indicates what types of data are to be expected by the computer, and what order they will be presented in. Some systems allow the order of the data to vary to some extent, but all impose some restriction.

We have experimented with such systems and found that they suffer from two serious defects. First, unexpected information always occurs in the sources, so no template can be adequate; data are always omitted. Secondly, with source data as complex as ours, any re-ordering of information imposes an unacceptable mental strain on those preparing the data.

Considerations such as these have led us to adopt a stringent design criterion for the computer input system. None of the information in the source was to be omitted. The entire text of the sources, in its original order was to be put into the computer.

It is necessary to impose some structure on the data put into the computer. If nothing but the original text were input, the range of analyzes which could be performed would be severely

restricted.

We intend to bring together references to persons from different sources, and we intend to automate this collation as much as possible. For the computer to assist in this operation at all it has to be able to identify the references to people in the sources. Theoretical linguists have spent a great deal of time and effort investigating this type of problem. Automatic parsing and understanding of natural language is one of the major goals of this research. However, practically usable systems which can do this do not exist yet. They belong in the same science fiction category as machines which can read handwriting.

So, something must be added to the text derived from the sources if this is to be processed automatically. These additions to the text must indicate which words refer to which people, plots of land etc. The additions must be made by human beings, who are good at parsing and understand natural language, for the benefit of computers which are bad at these activities.

The data model underlying our system

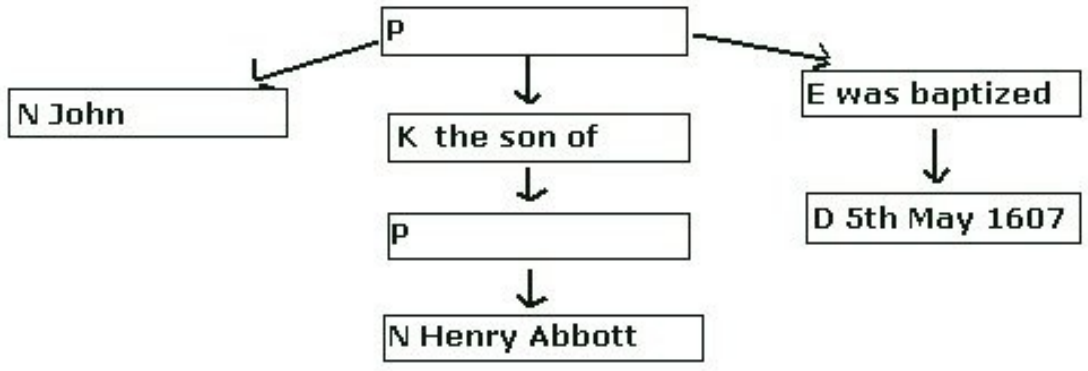
The system we have developed makes use of a relational data base management system. This is not the place to describe either relational databases in general or the details of our use of such systems. Space does not permit it. Date (1975) provides an account of these systems, and a commented bibliography of the subject. The relational model of data provides a theoretical frame-work within which it is possible to discuss the range of possible transformations of data. Use of a relational data base management system ensures that an adequate range of transformations will be available, and that the data can be analyzed in a wide variety of ways.

The way in which we model historical data within such a system can best be introduced by considering a simple example. Let us take a typical early parish register entry:-

"John the son of Henry Abbott was baptized 5th May 1607"

From our point of view this simple entry consists of words descriptive of a number of interrelated entities as follows:-
A person (1) who has a name (2), and who is involved in a kinship relation (3) with another person (4) who has a name (5). The first person in involved in an event (6) on a date (7).

The seven entities involved fall into five distinct categories: person, name, kinship relation, event and date. The model in the computer corresponding to this entry can be represented as follows:



Each box represents an entity. The words in the boxes are derived from the original text. The letters in the left hand sides of the boxes are category name abbreviations.

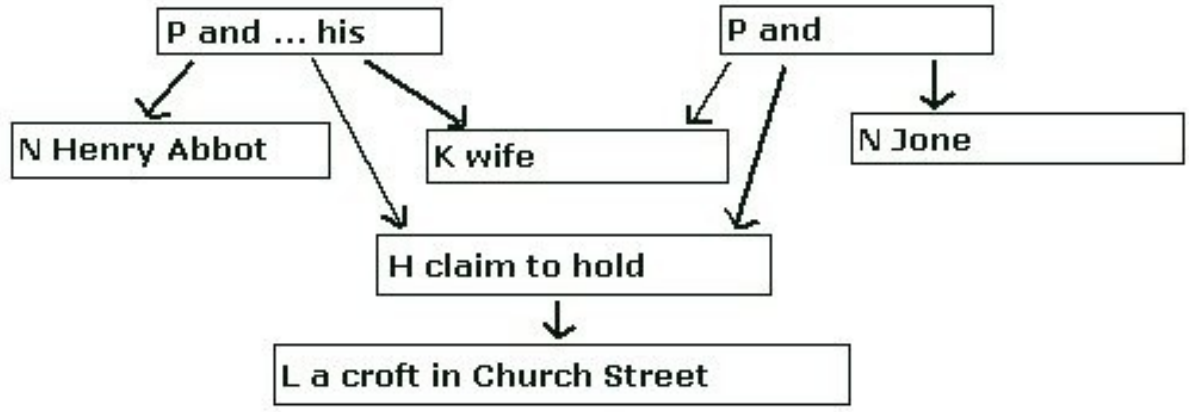
The arrows between boxes represent conceptual links between the entities. The meaning of such links is determined by categories of the entities involved. For example, a link between a person and a name implies that the name is the name of the person.

The data models arising from source text may be very complex indeed. A will may easily mention over a hundred entities. Space here only permits one further example, adapted from a rental, which will illustrate some further points.

The source text is:-

"Henry Abbot and Jone his wife claim, to hold a croft in Church Street"

and the corresponding data model looks like:-



'H' and 'L' are abbreviations for two new category types; landholding and description of a plot of land. Note that the word 'his', which grammatically is a back reference to Henry Abbot appears in the right box, and that the arrow leading from the box containing 'wife' is returned to this box. Also note that both the boxes are linked to the landholding box, thus capturing the meaning of the conjunction 'and'. The word 'and' itself, being part of the joint description of the two people, appears in both the person boxes.

These two examples introduce the principal features of the data model. The model is built up of: Words derived from the source text, Entities; Links. Each word is associated with one or more entities of which it is descriptive. Each entity is assigned to a category of entities, categories of entities are identified by conventional abbreviations. Links join pairs Of entities. The meaning of a link depends on the categories of entities involved.

The actual data model used is more complicated than this. Extra structure is stored to indicate the context of information, both its date and source, and words are supplied with sequence numbers to enable the source text to be reconstructed. Special provisions are made for numbers, dates and comments supplied at data preparation time.

The method of input

Data for input to the system must include, as well as a transcript of the source text, an indication of how the structure of entities and links is to be built, which categories the entities belong to, and which words describe each entity. All this information can be conveyed by adding brackets to the text. Let us consider again the parish register entry. The full annotated input to the computer system would be as follows:-

(P (N John) (K the son of (P (N Henry Abbott))) (E was baptized. (D 5th May 1607)))

There are seven pairs of brackets inserted, corresponding to the seven entities mentioned. The left hand bracket of each pair is immediately followed by the conventional abbreviation of the category name for the entity. The words associated with an entity are those which lie within the bracket pair, but outside any inner bracket pair. Each bracket pair except the outermost gives rise to a link between entities. For example, the event entity is linked to the date entity because the bracket pair for the date entity lies immediately within that for the event entity. The bracketing scheme, which indicates all the required structure, follows the phrase structure of the English fairly closely. This is no accident; in natural language, a phrase describing an entity is always embedded in its context in a way which permits it to be bracketed off. We have found that the system of bracketing works for a wide variety of sources, and that cases of difficulty are usually the result of grammatical errors in the original sources.

This simple example demonstrates the basic principles of the method. Nested phrases in the original text are marked off with nested brackets, each left hand bracket being labeled with the category name of the corresponding entity. However, it does not illustrate two important features of the scheme. The passage quoted above adapted from the rental illustrates these features. As prepared for computer input it would run:-

&(P (P *1 (N Henry Abbott)) and (P (N Jone) (K (1 his) wife)) (H do claim for hold (L a tenement in Church Street)) &)

The first new feature is the bracket pair &(P ... &) which encloses the entire passage. This is an example of the method of dealing with conjunctions. This bracket pair does not give rise to an entity. It performs the function of conjoining the two people mentioned. The word 'and', which lies directly inside the bracket pair is associated with both people, and links are generated from each of the two people to the landholding entity. The effect of this process may be seen by comparing the input with the diagram of the resulting data model given above.

The second new feature is represented by the bracket pair (1...) surrounding the word 'his'. This bracket pair does not give rise to a new entity in data model. Whenever a left hand bracket is followed by a number, the bracket pair is taken to represent a back reference to the last place in the text where an asterisk followed by the same number occurs. In this case the reference is to the person whose name in Henry Abbot. The effect is to add information derived from the new bracket pair to the pre-existing entity. In this case the word 'his' is added to Henry Abbot's entity, and the link from the kinship relation is made to point to this entity. Again. this can be seen in the diagram given above.

This back reference system allows for the computer to he told how to resolve the referents of pronouns and other definite descriptions. Resolution of such references is the most thorny problem for those who attempt automatic processing of natural language, although pronouns normally present no problem to human beings. The use of human as opposed to artificial intelligence to disambiguate such references is an essential feature of the system.

These, then, are the main features of the input system. Bracket pairs labeled with category types delimit the words associated with entities. Bracket pairs with '&' signs are used to form conjunctions of entities. Bracket pairs labeled with numbers are used in conjunction with asterisks to indicate back reference, and so resolve the referents of pronouns. Links between entities are implied by the way in which bracket pairs lie within one another.

Practical considerations.

Input for the system is usually prepared in two stages. First a machine readable transcript of the source is prepared. This transcript is in English with modernized spelling. Latin originals are translated. it would be possible to work the system with mixed languages but standardization of spelling is essential. The deciphering of words in archaic spelling is a typical pattern recognition problem. People are good at this sort of thing, and computers are not.

The raw transcripts are checked, and corrected using the standard text editor provided by our local computing service. This checking is facilitated by the use of a program which maintains a dictionary of all words in the input text, and checks new input against this dictionary. This program detects many mis-punches and spelling errors.

The addition of bracket pairs etc. is carried out with the help of a specially written text editing program. This program runs on a graphics mini-computer, and is driven by light-pen interaction. The program displays a portion of text on a screen. A bracket pair is inserted by pointing with the pen in turn at the positions for the left and right brackets, and then picking out a category name abbreviation and a bracket type from displayed menus. The program responds by redisplaying the text with the new bracket pair inserted. The process is then repeated.

With the help of this program the bracket pairs can be added fairly speedily and accurately. The speed varies between 100 and 400 lines per hour, according to the complexity of the source material. Some of the more complex sources, notably wills, require the addition of a great many brackets to indicate a very elaborate data model. However, as the bracketing closely follows the grammar of the text, it is not as difficult to do as might be expected. It is with this type of source material that the system shows its full advantages over more conventional methods. If the information from such a source were to be reordered to fit some predefined template, the mental effort involved would be considerable. With the system described here, the entities mentioned in the text are considered one at a time as they occur. No information has to be remembered, to be inserted at some other point in the input.

The range of possible applications of the system.

The input system and data model described here are but part of a larger proposed system. So far, we can get data into the computer structured according to the model. In collaboration with Mr. T.J. King, the authors are currently developing an enquiry language and information retrieval system to act on the data model. This system will provide the coded data required by multi-source record linkage algorithms, as well as automating many other indexing processes. Although this entire system is being developed with historical data in mind, it could in principle be applied in other fields.

The only part. of the system which is specific to the subject matter is the universe of categories of entities. We regard people, plots of land, kinship and other relations between people, ownership and other relations between people and plots of land, and dates as being the most significant entities described in our data. In consequence, our universe of categories of entity is based on this list. A

passage which includes but few references to such things will not have many brackets added to it, and will generate a simple data structure within the model.

Investigators with other interests and data would select differ basic categories of entity, and in consequence would bracket their data differently.

The distinctive feature of our data, around which the system is designed, is that the records pre-exist. The primary data collection process is out of our control; we must take the data as we find them. The system might well find application in other fields where data have already been recorded in unstructured natural language form without consideration of the requirements of automatic processing.

In cases where data are already highly structured, having been recorded on standard forms, or data have yet to be collected, the system does not work well. If data are to he collected they should be collected in a form to suit simpler automatic processing systems. If data pre-exist in highly structured form, more conventional methods of computer input work well, and the extra complexity of the system described here is redundant.

References

Date, C.J. *An Introduction to Database Systems* (Addison-Wesley 1975)
Macfarlane, Alan et al. *Reconstructing Historical Communities* (CUP 1977)
Wrigley E.A. (ed.) *Identifying People in the Past* (Edward Arnold 1973)