Computer Assisted Occupation Coding

Diane Bushnell, Office of Population Censuses and Surveys, UK

1. Introduction

Many papers have been written describing the advantages of using computer assisted personal interviewing (CAPI) compared to paper and pencil interviewing (PAPI). However, the potential of CAPI has not yet been exploited fully for the process of coding open-ended responses. Traditional methods are still widely used, i.e. the response is entered verbatim into the computer and coded manually later, either by the interviewer at home or by specialised coders back in the office.

In recent years, computer assisted coding has become more sophisticated, allowing the interviewer to search through a large database of responses and select an appropriate code during the interview. The usual benefits of computer assisted interviewing apply. The elimination of a separate data entry stage results in lower costs and a quicker turn-around of interview data, particularly where coding was previously carried out at HQ; data quality is improved due to a reduction in data entry and consistency errors and the data collected is more likely to be an accurate reflection of the interview as the data entry and editing stages are carried out in the presence of the data source (i.e. the respondent).

In Social Survey Division (SSD) of OPCS, we use computer assisted coding (CAC) in the interview for straightforward coding frames, such as country of birth and nationality, and most diary components of surveys are now coded by interviewers using CAC either at home or in the SSD Telephone Unit, e.g. items bought in last week, journey destinations. However, the coding of occupation is still carried out using the traditional methods.

We have been evaluating three systems for computer assisted occupation coding against a set of ideals for a model system. This paper describes our progress so far and a few preliminary findings, in terms of the technical abilities of the systems and their impact on data quality.

2. Occupation coding

In the United Kingdom, the Standard Occupational Classification (SOC) (OPCS, 1990) is the most widely used occupational coding scheme for social research. The SOC was developed for use on the 1991 Census of Population by OPCS, the Department of Employment and the Institute for Employment Research at the University of Warwick.

The classification consists of 371 occupational unit groups (SOC codes) which may be aggregated into minor and major groups. The system is hierarchical so, for example, unit group 270 (Librarians) is in minor group 27 (Librarians and Related Professionals), which is in major group 2 (Professional Occupations).

The major use of SOC in social research is for assigning individuals to various social class and socio-economic classifications.

In SSD the interviewers collect verbatim details on the job title, main duties and responsibilities of the job, qualifications required and industry, as well as asking specific questions about employment status (e.g. whether self-employed and the size of establishment). At home the interviewers go back into the questionnaire, review the details and select the SOC code which they think is most appropriate from a paper coding index. They then type the code into the Blaise instrument.

In theory, the allocation of a code to an occupation is quite straightforward: the interviewer simply looks up the job title in the SOC coding index. In practice, the procedure is much more complex. The index was published in 1990 (to tie in with the 1991 Census of Population) and contains around 21,500 entries. Even within five years many changes can occur in occupations: job titles become obsolete, new titles appear and job titles do not necessarily reflect the occupation of the job holder. Fashions may dictate that the same type of work is given a different title over time, e.g. Personnel Officers become Human Resources Managers, or whole new groups of workers are labelled with a title that had a very specific membership previously, e.g. a sudden increase in the number of able seamen was found to be due to a trend for staff working in MacDonalds to be titled 'Crew members'.

As well as problems with the index, people tend to vary the information they supply about their jobs according to the situation, so the same job could be described in many different ways. People may describe what they have been working on most recently, or may feel differently about their job at different times or may tailor the description according to the audience, e.g. I could describe myself as a civil servant, Senior Executive Officer, Senior Social Survey Officer, Statistician, Social Researcher etc.

Thus, interviewers (or coders) will need to make decisions based on all the information supplied, not just job title, in order to assign the relevant code. As with any process requiring human judgement this will give rise to variation over time and between coders. In the past, specialised coders were trained to apply standard rules rigorously; supervisors resolved queries and quality checks took place regularly. It is clearly more difficult to provide this amount of guidance on coding for face-to-face interviewers. Studies in SSD comparing field coding of occupation with office coding (Dodd 1985, Martin et al 1995) have shown that reliability is lower when coding is carried out by interviewers. In the most recent study, 8 interviewers, 5 office coders and 1 expert in occupation coding assigned SOC codes to 200 occupations and the reliability and accuracy of coding were examined. In this context, reliability is defined as the extent to which different coders assign the same code to the same case, and accuracy defined as the amount of agreement between the coders in the trial and the 'expert'. The average reliability of office coders was 0.82 and of field coders (interviewers) was 0.74 (a figure of 1 indicates perfect agreement on all cases). Accuracy was 0.80 for office coders and 0.77 for interviewers.

Although accuracy is not too badly affected by field coding the consistency of coding obviously suffers. On the plus side, the impact of individual coder bias, i.e. a systematic deviation in assigning codes compared to the other coders, is reduced by interviewer coding since the individual workloads are much smaller and spread over many more coders (the whole interviewing force compared to a handful of office coders).

In an attempt to address the difficulties with occupation coding we decided to investigate the use of a computer-aided coding system. In addition to the usual benefits of CAPI mentioned earlier, we hypothesised that computer assisted occupation coding would improve reliability by applying the standard coding rules more consistently than manual coding; accuracy would be increased by reducing the number of inappropriate codes presented to the coder; some of the burden on the interviewer would be removed as the task would be easier and the effect of coder bias would be reduced. It is important to note that although coder bias may decrease, the system itself may favour particular codes so a 'coding system' bias may be introduced.

In SSD, we began our investigations by considering three different coding strategies: using existing Blaise coding facilities, incorporating an external coding application into a Blaise questionnaire and using standalone occupation coding software.

Using Blaise was the obvious starting place as all our CAPI surveys are conducted using Blaise and we are already using the coding facilities for other coding frames. Tests were carried out with both Blaise 2.5 and Blaise III and the results are described in Section 3.2 below. The alternative strategies both involved using a specialised occupation coding application. In the UK, the most sophisticated software of this type is CASOC (Computer Assisted Standard Occupational Coding). The University of Warwick, with the University of Cambridge, began development of CASOC in 1986. CASOC is based on the OPCS Standard Occupational Classification and uses the same standard coding rules as OPCS. To investigate the second approach to coding occupation, using an external application from within Blaise, OPCS commissioned one of the authors of CASOC to produce a cut-down version which could be called as a user unit from Blaise. Progress so far is outlined in Section 3.2. The third option, using a standalone occupation coding package, although possibly the quickest and easiest to implement, was not considered compatible with our ultimate goal of integrating the occupation coding with the interview so no further work has been carried out in this area.

3. Computer Assisted Occupation Coding

3.1 Evaluation criteria

As a means of evaluating the coding procedures we devised a set of characteristics for a model coding system. The basic requirements were that it should use the job title to search through a coding index, present various code options, store the code for future use and then continue with the interview. The 'ideal' system should include as many of the following characteristics as possible.

The system should:

a. **Be easy to use**

Interviewers should need a minimum of special instructions.

b. **Display exact matches first**

If an exact match for the entered job title is found in the coding index then this match should always be displayed at the top of the suggestion list.

c. **Be at least as reliable and accurate as manual coding**

As one of the main aims of introducing computer assisted occupation coding is to improve reliability and accuracy of occupation coding this criterion is a very important one. We know that one of the problems with computer assisted coding in general is that interviewers favour codes at the top of the list and may select the top code even when it is clearly inappropriate. Thus it is important that, where an exact match does not exist in the index, the agreement between the code presented at the top of the suggestion list and the 'best' code is high.

d. **Be fast**

The process of assigning a code should be at least as fast as manual coding using a paper index.

e. **Be able to pick up information from the interview**

The system should be able to pick up information which has already been collected during the interview. This is useful for two reasons: coding may not always take place at the exact moment that the information is collected, e.g. initially the system will be tested by interviewers coding at home so information from the interview will be passed to the coding system at a later stage of the questionnaire; several pieces of information may be added together and used for coding so these would need to be picked up after all the data have been collected.

f. **Save the occupation code in the Blaise data file**

The code should be saved with the rest of the Blaise interview data for ease of analysis and to prevent losing data, making errors, etc.

g. **Have a simple mechanism for creating and updating the coding frame**

Although the SOC coding frame does not change very often it is essential that minor amendments can be made easily. The process of constructing the coding frame should be fairly straightforward and need no special technical expertise.

h. Be able to restrict search by auxiliary fields

It would be useful if the codes suggested could be restricted to a subsection of the coding frame by filtering on information other than the job title. For example, if the respondent was self-employed, then only valid codes for self-employed occupations would be displayed initially (although it would be necessary to have the option to extend the search to show all codes).

i. Include the possibility of displaying supplementary information on the occupation

It would also be useful if information pertinent to particular codes could be displayed. For example, synonyms for job titles (gaffer, foreman, supervisor etc.) or a summary of typical tasks carried out by job holders in those occupations.

j. Allow interviewers to take advantage of the hierarchical structure of the coding frame

Job titles will be assigned a code and this code will have a general name, known as the unit group heading. This heading can be used as a check that the correct code has been assigned. It might also be useful for the interviewers to step through the hierarchy, as a check on accuracy and also to increase awareness of the coding frame structure.

k. Include the possibility of automated and/or semi-automated coding

Automated coding can be defined as a system which assigns a code to the job title without any interaction with the coder. Semi-automated coding, in this context, is defined as a system where coding is automatic if there is one match with a goodness-of-fit above a certain cut-off point but where the coder assists with coding if the fit is not good enough. Both types of coding would obviously reduce the burden on interviewers and increase reliability but bias and accuracy may suffer unless the system is very good.

l. Include the option of excluding selected words from the matching process

To improve the performance of the matching algorithms it would be useful to exclude some words from the matching process. For example, if the job title contains the word 'and' this will introduce superfluous information into the match.

m. Be cost effective

The cost of developing and using a computer assisted coding system is obviously one of the key factors in our final decision making process. However, it is also one of the most difficult criterion to evaluate and so far, the cost implications of the various systems have not be explicitly analysed.

3.2 Evaluation

Coding systems were set up using Blaise 2.5, Blaise III and a combination of Blaise 2.5 and CASOC. Each of the systems was evaluated against the model characteristics: the results are summarised in Table 1. The occupation data used for the study on coding reliability was also used to test some aspects of reliability and accuracy for the systems.

Table 1 Three coding systems evaluated against 'ideal' criteria

Ideal Criteria	**Blaise 2.5**	**Blaise III**	**Blaise/ CASOC combination**
a. Easy to use	Y	Y	Y
b. Exact matches at top of list	N	N	Y
c. Reliability and accuracy at least as good as for manual coding	N	?	Y
d. Quick	Y	N	Y
e. Ability to pick up information from the interview	Y	N	Y
f. Occupation code saved in Blaise data file	Y	Y	Y
g. Easy to construct and update frame	Y	N	na
h. Ability to restrict search by auxiliary fields	N	N	Y*
i. Possibility of displaying supplementary information	N	Y	N
j. Ability to view information on levels of hierarchy	Y	N	Y*
k. Possibility of automated and/or semi-automated coding	N	N	Y*
l. Possibility of excluding selected words from matching process	N	N	Y
m. Cost effective	?	?	?

Y	Yes, system fulfils this criterion
N	No, system does not fulfil criterion
*	Not included in current specification for customised system but available in standalone version of CASOC
?	Not tested yet
na	Not applicable

Blaise 2.5

The three types of coding available in Blaise 2.5 (hierarchical, trigram and alphabetical) were used in combination in our test system. The occupational details were collected in the usual way. At the code question the coding window appeared in the bottom half of the screen. The question, displaying the occupational details required for coding, appeared in the top half of the screen. Initially, the top level of the SOC hierarchy was displayed in the coding window and the coder could step through the hierarchy or proceed by word searching. The job title entered previously was used for the search. Matches were displayed in the coding window and the coder could move up and down the list to select a code or could change the description to search for other matches. When a code was selected the unit group heading was displayed; the coder could accept this or continue searching until satisfied with the code selected. Finally, the code and description used for searching were saved in the data file and the coder returned to the Blaise interview.

The main advantages of this system were that it was easy to use and was familiar to the interviewers. The disadvantages were that only one field could be used in the coding process and no other information could be displayed to help the coder make the most appropriate decision, there was no automated or semi-automated coding and it was not possible to exclude words from the matching process. A significant problem was that the code at the top of the suggestion list was often not the most appropriate. Where an exact match to the job title exists in the coding index this was NOT always presented at the top of the list. Some further work is being done on changing the format of the index to reduce this problem. Reliability was not assessed but we did look at the extent to which the first code suggested agreed with that assigned by an expert coder (using the paper index). The data used for assessing the reliability and accuracy of field and office coders was copied into the coding system and trigram matches found for the 200 job titles. The code at the top of the search list was the same as the expert's code in 43.5% of cases (87 out of 200). Again, it may be possible to improve this figure by changing the format of the entries and removing punctuation etc, but as it stands, this figure is unacceptably low.

The principal costs for this coding system will be incurred in development of the coding frame and piloting.

Blaise III

At present, the coding module of Blaise III is still under development so all the following comments should be regarded in that light.

The basic coding functions of Blaise 2.5 are available in Blaise III. It is possible to carry out coding by means of a hierarchical, trigram or alphabetical search although at the time of testing it was not possible to integrate hierarchical with word match searching. A new feature is the ability to include more than one search field in the coding index. The coding scheme tested involved only trigram and alphabetic search procedures. It was not possible to pick up text previously entered so the job title was entered directly onto the coding screen. Trigram searching was set as the default with the option to switch to alphabetic searching. After searching the database index entries with a 'good' match were displayed and the coder could move up and down the list and change the search description as for Blaise 2.5. When the code was selected it was written to the Blaise data file, along with the description held with the code in the coding index (not the description entered by the coder).

Not many of the ideal criteria were met by the Blaise III system (see Table 1 for a summary). The poor performance of this system was partly due to the incomplete development of the coding module in Blaise III and more of the criteria will be met as the program is refined.

The only real advantage of this system over the Blaise 2.5 system was that it was possible to have more than two fields in the coding index and any of these fields could be used for searching. For instance, industry could be used as a search field or it could just be displayed on the coding screen with the other information.

The process of creating the coding frame has changed significantly in this version of Blaise and, even allowing for learning time, took much longer than in Blaise 2.5. Manipula was used to convert the text coding index into a Blaise datamodel and 21,500 codes and descriptions took almost eight hours to convert. In most circumstances the coding frame would only need to be converted once so this should not be a critical constraint. However, it does mean that it may take a while to make relatively minor changes to the frame. The size of the subsequent datamodel and auxiliary files required for coding was about 12MB compared to under 2MB in Blaise 2.5. The time delay before trigram matches appear on the coding screen has also substantially increased so that there is a noticeable gap between typing the job title and the matches appearing on the screen. As the coding window fills most of the screen it obscures the occupational details displayed but presumably it will be possible to customise the location of the window in later versions. It should be possible for the Blaise team to fix all these problems with further work.

As in Blaise 2.5 there is still no facility for automated or semi-automated coding, nor the ability to restrict the search to subsets of the coding index. The building blocks for these functions seem to be in place already so it would not be too difficult to provide them, if there was enough demand by users.

As there are obviously some problems to be sorted out with this version of Blaise no tests of reliability or accuracy have been carried out.

As with the Blaise 2.5 option, the major expenditure will be on creating a good coding frame and on piloting the system.

CASOC in the Blaise instrument

It is not possible to call external programs from within Blaise III yet so we were only able to test this option using Blaise 2.5. A version of CASOC was written in Pascal and compiled with the Blaise instrument as a user defined question type. At present the system is still under development so it is only possible here to describe the specification for the system. The occupation details will be collected during the interview as usual. At the coding question the customised version of CASOC will take over and use the job title to search through its database. The appropriate codes and descriptions will be presented on the screen and the coder will be able to move around as if using Blaise. In fact, we have decided to make the interface with CASOC as similar to using Blaise 2.5 coding as possible. This means that the interviewers will be able to use the system with a minimum of training above that normally required for Blaise 2.5. When the code has been selected it will be saved in the Blaise data file and the interview will continue as normal.

As well as having the benefit of looking and behaving like Blaise, the system will have all the advantages of CASOC. The CASOC coding system employs a weighting system to arrive at the 'most likely' code as well as a word matching mechanism. Where job titles are similar the most common occupation will be first, e.g. Secretary/typist will be listed before Secretary of State. In the standalone version of CASOC it is possible to incorporate extra information into the search, for example, employment status or industry, and to move around the hierarchy of the coding frame, but these facilities have not been included in the specification for the current customised system. The frame itself is prepared by the authors of CASOC so that changes will be carried out automatically when appropriate. Potential disadvantages of the system are the costs of development and of using CASOC in tandem with Blaise, also the lack of control inherent in any system which is not developed in-house.

Due to the incomplete status of this project there are obviously no figures for the reliability or accuracy of this method of coding yet although some tests have been carried out using standalone CASOC. Campanelli et al carried out two separate experiments where occupations were coded using CASOC. In the first, five coders experienced in coding occupation on the Census assigned occupation codes to 401 cases. Three occupation experts coded the same information and their results compared to the professional coders' as a means of testing the accuracy of coding. Average reliability for the coders was 0.78 and agreement with the expert occupation coders ranged between 0.69 and 0.79.

In the second study, coders assigned codes to 322 occupation titles. Coding was carried out using the traditional technique first (consulting the paper index) and after a suitable interval, the process was repeated using CASOC. The occupation experts from the first study also coded information in this second study. There was little difference in reliability for the two methods (around 0.80) and agreement between the coders and experts was slightly higher when using CASOC than using the paper-based index (accuracy ranged from 0.78 to 0.85).

The accuracy figures for CASOC found from these studies may not be as high as they appear as two of the experts used for assessing accuracy were the authors of the CASOC system. Thus, one might expect good agreement between CASOC and its creators.

The reliability statistics are lower than those achieved by office coders using manual coding in the study conducted by SSD (see Section 2) (0.82) but higher than those for interviewers (0.74). The accuracy figures seem similar for all studies.

From these results, we might infer that using CASOC for occupation coding will improve reliability, compared to manual coding by interviewers, and that accuracy of coding will not be reduced.

To test how good the first suggested code was when using CASOC, the 200 cases from our reliability study were coded in fully automated mode. 69% of the cases (138 out of 200) were the same as those assigned by the expert coder. This is much better than the equivalent test carried out using Blaise 2.5 where only 43.5% of the codes listed first agreed with the expert coder's.

4. Summary

From the investigations into computer assisted occupation coding in SSD to date, a system which combines Blaise and CASOC seems to offer the most promising way forward.

The system using Blaise III met very few of the criteria we had devised for a model coding system but as Blaise develops some of the problems should be resolved. Blaise 2.5 fared better and was an attractive option in its ease of use, requiring little special training for interviewers. The main drawback of this system was the lack of a suitable coding frame. It was clear that a straightforward conversion of the paper coding index to computer was not acceptable: although it was quicker to look up the code electronically than using the paper index, there was a greater chance that the codes presented to the interviewer would be totally inappropriate. This would result in a decrease in accuracy and possibly an increase in bias (as some codes will appear more frequently than others).

Combining a specialised coding application (CASOC) with Blaise met many of the essential criteria for a good system as they were deliberately built into the specification or existed in the chosen software already. However, no conclusions can be drawn yet about the method as it is still under development. Apart from the possibility that this solution will not work at all, the main disadvantage is that it is only available under Blaise 2.5 at the moment. We hope that Blaise III will allow the inclusion of user functions and procedures soon. Studies have found that reliability and accuracy of occupation coding are as good or slightly better for coders using CASOC compared to traditional manual coding techniques. We will be carrying out our own tests when the combined Blaise 2.5/CASOC system is ready.

Over the next year, we intend to pursue the use of the combined system and to keep a close eye on developments in Blaise III. We will also be carrying out some qualitative work with interviewers on the process of coding occupation with the intention of tailoring our procedures to improve the quality of coding.

References

Campanelli P C, Thomson K, Moon N, Staples T (1995, forthcoming) *The Quality of Occupational Coding in the UK* In Survey Measurement and Process Quality, ed. L Lyberg, P Biemer, M Collins, E DeLeeuw, C Dippo, N Schwarz, D Trewin

Dodd T (1985) *An Assessment of the Efficiency of the Coding of Occupation and Industry by Interviewers* New Methodology Series No. NM14, London, OPCS

Martin J, Bushnell D, Campanelli P and Thomas R (1995) *A Comparison of Interviewer and Office Coding of Occupations* Joint Proceedings of ASA/AAPOR: Section of Survey Methods Research, American Statistical Association, Washington, DC.