Chapter 7

Data Coding and Entry – Lessons Learned

Pércsich Richárd

Introduction

In this chapter we give an overview of the process of coding and entry of the 1999 pilot test data for the English examination reform. The most important design principles and conventions of data management will be dealt with. The primary focus of the chapter will be to describe the stages of the procedure. Some of the problems arising from the lack of a standard set of criteria for quality control will also be touched upon briefly. If data analysis is to be efficient, reliable and believable, then it is essential to ensure that the raw data are input in a suitable form for analysis. This means giving careful consideration at the stage of the design of the test booklets and other data collection instruments to the needs of data processing, as we briefly recounted in Chapter 4.

Coding – rationale for item level recording of test takers' answers

In general terms, the process of coding means the transcription of answers recorded in a research instrument into machine- (computer-) readable format. The instrument is usually a series of questions in written format but it is possible to collect data using verbal instruments as well. In the educational context these may be written and oral examinations.

In language testing the reasons that the answers of a test paper are usually coded are varied. First, we are usually dealing with a large set of data, and it is not feasible to assess candidates' responses manually, that is to employ people to mark answers and calculate and report scores. This may be possible within a school or even a region but is unlikely in the case of an exam which certifies thousands of candidates.

Furthermore, test takers' answers often need to remain documented for long periods of time, for the purpose of reference or further study. National and international exam boards usually store data related to their test takers' performance. This allows for repeated analysis and gives a basis for further test development.

Machine-readable data is the most important raw material for test development. Once a set of data is objectively and reliably coded and entered into the computer, it is the basis for empirical research aimed at enhancing the actual tasks of a test paper and seeking ways of improving the methods of measuring a given skill or knowledge area. Pre-testing and piloting produce empirical data to allow the correction and revision of items and tasks, but using test results after the exam has been taken in order to design new tasks is also a sound alternative. These were important considerations in the 1999 pilot.

The instruments – means of collecting data

From the point of view of data management an important aspect of any research is to control the way data is collected, stored, retrieved and analysed. The research instruments should be a reliable source of information, make it possible to access the results and even repeat the examination if needed.

When test takers give answers to the questions in a language test they are required

(where appropriate) to formulate their answers in such a way that it is possible to mark them as right or wrong. In the case of multiple-choice items, a computer can score responses as correct or incorrect by comparing the student response to the key. In the case of other test types, as used in the April pilots, such as short-answer questions, cloze and gap-filling, and so on, humans first have to decide which responses are correct and which are unacceptable, and then code the responses accordingly. This tends to work well with receptive skills (listening and reading), while productive skills (writing and speaking) still remain problem areas (as detailed in Chapters 6, 10 and 11).

This chapter will concentrate on the management of data from the April 1999 Written Examination pilots. The results of the Writing and Speaking tests were handled differently, because of their nature and the small amount of data to be analysed. In effect, the scores given by human markers to performances on the Writing and Speaking tests were simply entered into an SPSS-readable format, and analysed using the standard procedures reported in Chapters 8, 10 and 11.

As we saw in Chapter 4, the instruments of the 1999 Written pilot were six test booklets and a student questionnaire.

Questionnaires

Most of the answers to the questionnaire items were either numerals (e.g. age, number of lessons), or characters ('x'-es) which could be entered without coding. Data about candidates' previous experience with the tasks ('Have you done a similar task before?') were recorded as Yes/ No, and their attitude to the tasks (perceived difficulty ranging from 'very easy' to 'very difficult') were coded on a scale from 1 to 5.

Some of the answers required some form of coding, usually abbreviation (e.g. m for male, f for female) – this was done by the data entry staff after a short technical briefing. The questionnaires did not contain open-ended questions.

This information was recorded and stored together with responses to items in order to be able to calculate correlations between test scores and various background factors. Because the analysis of background data was related to the actual test scores the proper analysis of the piloting data required many times more work than would have been necessary in the case of an examination. However, the analysis of the data provided by the questionnaire together with the results of the six different test instruments resulted in valuable information about the sample population.

Test booklets, task types and booklet design conventions

Since different language areas were targeted the test booklets were of different content and length. The two Listening booklets (No. 1 and 2) were fairly short (25-27 items) while the Reading and Use of English booklets contained 46 to 90 items. Because of these structural differences it was not possible to handle the data as one data base – six separate files were created for the six booklets.

Table 7.1: Test booklets

Booklet No.	Skill(s) tested	Number of items
1	Listening	25
2	Listening	27
3	Reading and Use of English	46
4	Reading and Use of English	58
5	Reading and Use of English	80
6	Reading and Use of English	90

Booklets were designed in such a way that it was later possible to do coding (and data entry) in a fairly simple way. Answer columns were provided where the task type made it possible or where it was reasonable to ask the test taker to actually code their answers themselves by putting the code (letter or number) of the answer in appropriate cells.

The reason that not all tasks were coded the same way (by the same process) was due to the nature of the tasks: the different mechanics of tasks did not allow for a uniform coding procedure to be applied. Table 7.2 gives examples of the variety of task types and related coding procedures.

Table 7.2: Different task types and coding procedures

Procedure	Task type or format	Answer type	Interpretation of answers	Human coding
Giving one-word answer	Gap-filling	1 word	Match key	Yes
Giving short answer	Open ended short answer	1 or more word(s)	Match key OR interpret + decide	Yes
Circling letter of correct answer in list	Multiple choice	Visual information	Match key	Yes
Marking letter of correct answer on map	Information transfer	Visual information	Analyse + match key	Yes
Writing letter of correct answer in answer column	Any	Letter	-	No

As an important testing principle the tasks were intended to be authentic in that, in the case of newspaper articles for instance, texts were supposed to be organised in much the same way as they appear in real life. It was then only a secondary consideration to provide enough space for an answer column – candidates were expected to concentrate on the task rather than make the work of markers easier.

This is why in the case of a Use of English task (Booklet 3, Part 6) the problem of coding was not solved: candidates were instructed to put a box around problematic words in the text, while markers were supposed to code each answer (33 items) in boxes on the bottom of the page. This method proved to be too tedious to be applied to hundreds of booklets, and the chances of human error were high. Therefore, no item analysis of this task was conducted, and only the total scores were recorded.

In addition to the importance of test authenticity, a major concern was to pilot as many types of tasks as possible. The test booklets thus contained tasks that fell into two major categories: selected response (as in multiple-choice) or limited production (as in gap-filling). Some of the most important task types were:

- *Short answer*: an open-ended question to which the test taker was expected to give a short answer, possibly one or two words would be adequate for an appropriate answer.
- *Multiple choice:* classic 4-option multiple-choice items were used in some tasks (the number of 3-option items was limited).
- *Multiple matching:* both questions and answers appear as options in a two-column format and the test taker is expected to match them.
- *Sequencing*: the test taker was supposed to find the correct order of a chain of events. This task type was expected to present problems of scoring, since counting exact matches did not give credit to partially correct answers. New methods of scoring this type of task were tried out (see Alderson, Pércsich and Szabó, 2000).
- *Information transfer:* where the test taker was expected to mark the answer on a visual (map).

Booklets thus contained a variety of tasks which had to be marked and coded. As in any large-scale assessment project the aim of coding was to make sure that the answers of the respondents could be easily interpreted by the data entry staff who were not trained to assess the actual answers. In fact, most did not speak English at all.

Markers, therefore, were expected to evaluate the answers test takers gave and put an appropriate code in an answer column after each task. Responses were coded as follows:

Correct answer	1
Incorrect answer	0
Omitted item	9
Not reached item	9

Although it is possible to use a more detailed system of codes, this procedure seemed to be adequate for both test development and calculating scores. While some statistical programmes allow for the distinction between omitting and not reaching an item, coding was simplified to the 9 code in both cases, as distinguishing between the two was expected to result in a reasonable amount of errors on the markers' part, while the possible advantages in item analysis would not have been worth the effort.

Unfortunately, the software used to analyse the items treats 0s as blanks, rather than as incorrect responses. We therefore had to use the 'search and replace' function to change all 0s to 2s. In future, markers should write 2s beside wrong answers in the students' test booklets, instead of 0s.

The marking session – increasing the reliability of assessment

Since most of the booklets contained tasks that required the test taker to give short answers to questions, or fill gaps in a text, some tasks needed to be coded. The first version of the marking key was not expected to be fully correct, thus a marking session was organised with the aim of revising the key and marking tasks that needed to be coded before data entry. An account of this appeared in Chapter 6.

The marking session proved to be valuable for three reasons:

- marking became more objective and therefore reliable
- the dynamic, interactive forum made work efficient – cheaper and quicker than individual work
- markers were able to negotiate their findings with the rest of the group and probably gained more professional insights than they would have working alone.

Data entry

The set of coded data was entered into the computer to form a data base. The structure of data bases may vary greatly but they have one common organising principle: it is usually the candidate (individual) whose unique identifier (name, or more likely, number) is the identifier of a *record* which may contain hundreds of *fields* (entries) including the actual answers to test items as well as background data/ questionnaire responses.

Microsoft EXCEL, a well-known commercially available spreadsheet program, was used for data entry. The reason for using this particular computer program was twofold. Since the program is commonly used in offices and at home it was easy to find people who had the necessary skills and experience to operate it. Secondly, the project did not own a program other than SPSS – a software package used for statistical analysis that requires advanced computer skills. Training people to use SPSS for data entry was less practical than using EXCEL and then transforming the data into SPSS or other formats for statistical analysis.

EXCEL allows for 256 cells to be used in one row of the spreadsheet, and so altogether 256 test and questionnaire items can be stored for one candidate. Alternative software was not explored at the piloting stage, as the available resources were expected, and later proved, to be sufficient for the purpose. However, a large scale examination – a national examination – may require a carefully-designed, tailor-made application that can handle much larger data sets.

The data structure and the related data entry format and procedure are usually designed before candidates actually take the test, preferably before the test papers are printed. It is essential to identify all necessary items of data in a very early preparatory phase, since research questions can only be answered by the data coming from adequate questions and appropriately coded answers of test takers.

Data entry needs to be done according to a straightforward procedure, as the data entry staff is not (should not be) expected to process the data in any way. Brief training regarding practical questions and the possibility of consultancy with a supervisor is also needed in an attempt to increase the effectiveness of the entry.

In the case of the April pilots, however, data entry was not designed in detail until the tests had been taken and booklets had been collected. This resulted in management problems – most importantly a delay in work schedules. Some problems of co-ordination of data entry also occurred as booklets were shared among too many people doing data entry which caused confusion and required the re-run of statistical software.

Data transfer – compatibility of assessment software

Table 7.3 shows that data analysis was fairly complex. Five different computer programs were used for various types of analyses which were often based on each other.

Table 7.3: Computer programs used for data analysis

Software	Purpose	Preferred data type
Microsoft Excel	Data entry and management	Short alphanumeric
MicroCAT Iteman	Classical item analysis[+]	ASCII
BigSteps	Rasch analysis[+]	ASCII
SPSS	Descriptive statistics, correlations, analyses of variance / tests of significance	Short alphanumeric, Data imported from Excel
Microsoft Word	Editing ASCII files, Reporting research findings	Text, formatted tables of data, graphs

+ These terms are clarified and results presented in Chapters 8, 10-15

Problems in data entry and data transfer usually caused a chain of events: once an error had been identified in one of a number of consecutive steps of data analysis, the entire process had to be started all over again. Indeed, trivial problems such as improper handling of data files as well as incompatibility of software used for analysis caused a great deal of frustration at times.

Tracing and correcting errors

The reason that the final data base needs careful error checking is that a data set does not look problematic in itself, nor does it necessarily tend to behave problematically. It may be used for a variety of analyses but if the user lacks a critical perspective – an alertness to identify odd correlations for instance – errors in the data may not be identified, and he may prove or discard a hypothesis incorrectly, supported by the illusion of using state-of-the-art statistical software.

Systematic and non-systematic errors in data entry may both have an effect on the results. These may be categorised as follows:

Systematic – error in all related items:

Incorrect key (due to printing error, etc.)
Incorrect application of key by human marker
Failure of computer program to handle some type(s) of data correctly.

Non-systematic – a casual mistake in some items:

Incorrect reading of key by human marker
Incorrect reading of code by data entry person
Incorrect entry of item (typing error) by data entry person.

When candidates are assessed in an exam as individuals, it is crucial for their answers to be entered into the computer correctly since their career may be at stake. This calls for attention to non-systematic errors in the data. Since a valid and reliable method of checking non-systematic errors was not applied in our case, it would not be possible to use the results of the pilot for certification of students within an acceptable margin of error. This, of course, was not the aim of the pilot. However, critical inspection of the resulting data and analyses did not reveal any cause for concern.

All possibilities for systematic error were checked and ruled out during coding and before data entry began. This made it possible to do a large-scale analysis of test results, including the correction and improving of items as well as item calibration.

Conclusion

The most important findings of this experience were that data management is most efficient when planned together with the research as a whole, and any deviation from the stages designed causes further problems whose correction takes considerable time and effort. While it is reasonable for data analysis to proceed according to a list of steps, a detailed documentation of the deviations gives an opportunity to revise the process later on.