Journal of Official Statistics, Vol. 15, No. 1, 1999, pp. 67–85

A New Method for Studying Verbal Interactions in Survey Interviews

Wil Dijkstra[1]

An integrated system to study question–answer sequences in survey interviews is presented. This system consists of (1) a multivariate coding scheme for coding the verbal utterances of interviewer and respondent; (2) a computer program to facilitate the transcription and coding of these verbal utterances; (3) a computer program offering a wide variety of ways to analyse the coded sequences.

Key words: interaction coding; question–answer sequence; sequence analysis.

1. Introduction

The study of interactions in survey-interviews is concerned with the course of verbal utterances between interviewer and respondent. Sequences of such acts are called **q**uestion–**a**nswer sequences, or **Q–A** sequences for short. They usually consist of questions, answers, requests for repetition or elucidation, and so on;

In order to study **Q–A** sequences three problems must be attacked.

– Firstly, an appropriate coding scheme should be available to represent the course of interactions in various types of survey interviews.
– Secondly, a convenient procedure is required to apply this coding system to verbal interactions in an efficient and reliable way.
– Thirdly, convenient, analysing techniques are needed for the coded **Q–A** sequences.

An integrated system was developed in order to meet these three requirements. It will be shown that this system greatly facilitates research into verbal interactions in survey interviews.

2. Interaction Coding in Survey Interviews

Systematic observation of what occurs in interviews has evolved from social and clinical psychology. Matarazzo and his co-workers (e.g., Matarazzo, Saslow, and Matarazzo, 1956) used the Interaction Chronograph devised by Chapple (1949) and later the Interaction Recorder (Wiens, Matarazzo, and Saslow, 1965) for studying paralinguistic speech (reaction time latencies, interruptions, speech durations, eh-saying, etc.) in both initial clinical and applicant interviews. Both devices depended on observers who were supposed

[1] Department of Social Research Methodology, SCW, Vrije Universiteit, De Boelelaan 1091 c, 1081 HV Amsterdam, The Netherlands. E-mail: w.dijkstra@scw.vu.nl.

to push buttons at the right moment. The research of Matarazzo and his co-workers typically shows that systematic variations of such speech characteristics by the interviewer are reflected by the respondent (for an overview, see Matarazzo and Wiens 1972). Studies of paralinguistic speech effects in survey settings are rare (some examples are: Marquis, Cannel, Laurent 1972; Barath and Cannell 1976; Blair 1978; Koomen and Dijkstra 1975.)

Most studies of interviewer and respondent behaviour in survey interviews concern verbal speech. The systematic study of behaviour in interviews became feasible as soon as taperecorders became available (e.g., Belson 1967, 1969; Wilcox 1963.) Cannell and his co-workers were the first to devise an instrument to systematically study this behaviour. Their manual for coding and analysing interviewer behaviour (Cannell, Lawson, Hausser 1975) includes a coding scheme, examples, and a computer program to calculate frequencies and transition frequencies of observed behaviour. This instrument was designed primarily as a technique for evaluating the performance of interviewers. The coding scheme comprises 39 codes for interviewer behaviour only. Codes for respondent behaviour are not presented. Codes are grouped into nine broad categories. Each code consists of two figures, the first one signifying this broad category, the second one a further specification. For example, category 1 concerns ''question read correctly,'' and can be subcategorised into 11 (''reads question exactly as printed in questionnaire'') and 12 (''reads question making minor modifications, but does not alter frame of reference''). A simplified version of the coding scheme can be obtained by only using the first figure.

The coding scheme of Cannell et al. can easily be extended to account for respondent behaviour. This was especially thought useful for evaluating (or pretesting) the questionnaire: if respondents find difficulty with particular questions, then their behaviour will show it. Morton-Williams (1979) used a coding scheme of 29 codes, and included respondent behaviour like ''asked for repetition of question'' or ''not answered adequately''. Blair (1978) distinguished 58 behavioural categories, and included respondent behaviour as well. A scheme developed by Prüfer and Rexroth (1985) for evaluating question quality numbered no fewer than 87 codes, of which 28 described respondent behaviour. Oksenberg, Cannell, and Kalton (1991) not only sought to identify problematic questions by coding respondent behaviour, but also to identify the nature and cause of the problem in order to obtain information on how to improve problematic questions. Brenner (1982, 1985) used a coding scheme based on ''rule-breaching'' and ''rule-following'' actions by the interviewer. Rule-breaching actions are for example ''directive probing based on respondent's information,'' or ''question completely altered.'' Rule-following actions are ''question asked as required'' or ''interviewer clarifies adequately.'' In addition a number of respondent actions were distinguished, for example, ''respondent answers adequately'' or ''respondent's information is irrelevant.'' Brenner does not provide codes for these categories, apparently because he did all the coding work himself. Brenner was interested primarily in the course of actions: how does a particular action by the interviewer affect the respondent's action, and how does this respondent's action affect the interviewer's behaviour in turn.

More recently, behaviour coding has been used for studying the cognitive processes involved in answering questions (Fowler and Cannell 1995; Sudman, Bradburn, Schwarz

1996; Tucker 1997). This approach applies protocol-analysis of problem-solving tasks (Ericsson and Simon 1993) to interview-protocols, adding the intricacies of the interactional process between interviewer and respondent (Schaeffer and Maynard 1996).

In sum, observation techniques serve the following purposes: monitoring interviewer performance and identifying questions that may cause problems to both interviewer and respondent, studying the course of the social interaction between interviewer and respondent, and the cognitive processes involved. In addition to the literature cited above, an excellent overview of these uses is provided by Cannell and Oksenberg (1988). Although this overview primarily concerns behaviour during telephone interviews, it covers face-to-face interviews as well.

3. A Multivariate Coding Scheme

The overview of coding schemes discussed in the previous section is far from exhaustive. Researchers tend to adjust existing coding schemes (usually based on the instrument developed by Cannell et al. 1975), or to develop their own scheme from scratch (like Brenner), depending on particular research questions to be answered. Although most coding schemes have some categories in common (like ''Question read as worded'' or ''Answer inadequate for coding,'') most categories differ between coding schemes. This diversity makes it difficult to compare the results of different studies. Moreover, especially if one wants to study the interaction between interviewer and respondent, it is necessary to cover all behaviours that potentially may affect the outcome of the interaction (the eventual answer). Hence, the number of categories may be quite large. But even in coding schemes with a relatively large number of categories, particular categories are usually missing that clearly may affect the interaction as well as the eventual answer. For example, notwithstanding the rule that respondents should be interviewed alone, Hartmann (1992) reports that this requirement is usually met only in about two thirds of all interviews in social science research. Nevertheless, categories accounting for the bystander's behaviour are typically absent. As another example, a respondent may ask the interviewer how he or she would answer a particular question. Extending code schemes with such infrequent but significant behaviours may easily lead to an unmanageable number of categories and codes.

Both problems, the comparability between different observation studies and a large number of categories if one wants to account for a large variety of behaviour, were attacked by constructing a multivariate coding scheme. Essentially the idea is that each (verbal) act by interviewer and respondent is coded on a number of coding variables. Each coding variable consists of a limited number of categories. The coding variables range from general (describing broad categories) to more specific. General coding variables are intended to be used for a large range of different studies, thus making comparisons between different studies possible. Specific coding variables are intended to answer particular research questions and can differ for different studies. On the face of it this approach may seem fairly complicated; instead of making one decision per verbal utterance to be coded, a coder has to make a number of decisions, namely one for each coding variable. We will show that the actual coding can be done in a fast, easy and reliable way.

We will discuss five coding variables to be used for standardised survey interviews, and yet another example of a sixth coding variable, to answer specific research questions[2]. In addition we will show that the first four coding variables can also be used for non-interview interactions, i.e., the introduction to an interview, to describe the process of persuading persons to participate in an interview.

3.1. The actor

The first and most general coding variable designates the speaker, or ACTOR. That can be the interviewer (coded as I), the respondent (R) or a third person (T). Should one wish to discern between particular third persons, for example to study the impact of their presence, codes for partner (P) or child (C), etc. can be added.

3.2. Information exchange

The second variable is also very general and can be used for most conversations. It describes the type of information exchange, and is called EXCHANGE. We discern between question (Q), answer (A), perception (P), request (R), comment (C) and detour (D). A question (Q) is a query to another actor to obtain information. An answer (A) signifies that information is provided, although not necessarily as a response to a question.

The category perception (P) covers behaviour that signifies that the other actor's utterance is received or understood, for example by repeating that utterance. A request (R) signifies that an utterance is not understood: an actor may request that the other actor repeats or elucidates an utterance.

An actor can also comment (C) on another one's utterance. Examples are ''That's a difficult question,'' or ''That's useful information.'' Finally, verbal utterances can depart temporarily from the main stream of information exchange within a Q–A sequence. Such detours (D) can be related to the interview, like ''How many questions are there still for me to answer,'' or they may not be related to the interview at all: ''Would you like a cup of tea?''

3.3. Relevancy or distance

The third variable concerns the relevancy of the verbal utterance. This coding variable is called DISTANCE and describes the ''distance'' between a verbal utterance and the purpose of the interaction. In a standardised survey interview the purpose is to obtain an answer to a question from the questionnaire. If the interviewer poses a question from the questionnaire, the distance is zero, for the behaviour is directly related to this purpose. Similarly, an answer of the respondent to such a question, for example the choice of a response alternative, also has a distance of zero.

If the interviewer asks the respondent to motivate such a choice, the distance of that question is 1. Also the motivation by the respondent itself has a distance code of 1. Generally ''distance 1'' concerns direct elaborations or motivations of a ''distance 0'' answer, irrespective of whether such an answer has already been given or not. The respondent may subsequently digress from the topic, for instance by giving examples or giving opinions that are related to the topic only indirectly. This is coded distance 2.

[2] A paper describing the coding system in more detail and with lots of examples, can be obtained from the author.

Distance 3 is used for utterances that are completely irrelevant to the scripted question. A ''distance 3'' utterance should be well distinguished from a ''detour.'' A ''distance 3'' utterance, although irrelevant with respect to the scripted question, ''builds'' on previous actions with distance codes of 0, 1 or 2.

Sometimes utterances refer to previous sequences (''You said earlier ...''), or to future sequences (''We will ask some questions about that later on''). These utterances are coded 'B' ('backward') or 'F' ('forward'), respectively.

3.4. An illustration

Example 1 illustrates these first three coding variables. The Q–A sequence (like all other examples) came from telephone interviews conducted by the University of Wisconsin.

The Q–A sequence begins with the interviewer (I) introducing a scripted question. Such introductions to the question proper are coded 'Q' for ''question.'' Utterances two and four concern the scripted question itself. The third utterance (''Yes'') is an answer by the respondent to the scripted question; clearly the answer is incorrect as it is based on only the first part of the question. After finishing the question, the respondent answers the question again. The utterance ''but my car'' is clearly a direct elaboration of this answer and is coded '1' for distance. Finally, the interviewer shows that he or she has perceived the answer.

3.5. Specification of information exchange

The fourth coding variable is a further SPECIFICATION of the EXCHANGE categories. The categories to be discerned depend partly on the category of the exchange variable.

Example 1:

ACTOR	EXCHANGE	DISTANCE	Verbal utterance	Description
I	Q	Ø	I: And now we'd like to ask you a couple of questions about experiences you may have had as a victim of crime	introduction to question from questionnaire
I	Q	Ø	I: During the past twelve months did anyone break into or somehow	I reads scripted question
R	A	Ø	R: Yes	R gives (preliminary) answer
I	Q	Ø	I: illegally get into your home and steal something	I finishes scripted question
R	A	Ø	R: uhm not my home	R gives answer
R	A	1	R: but my car	R elaborates
I	P	Ø	I: Okay so it was not your home	I signifies that answer is perceived

If an utterance is coded as a ''question,'' we distinguish between closed questions ('C'), yes/no questions ('Y'), open questions ('O'), and presenting response alternatives ('A'). An introduction preceding a question is coded 'I'. Explanations of the meaning of a question, are coded 'M'. A closed question is not necessarily a question accompanied by response alternatives. Essentially a closed question is a question asking the respondent to take a position on a unidimensional variable, for example ''How satisfied are you with your house?''; ''What's your age?'' An open question does not specify such a variable, for example: ''Why are you dissatisfied?''; ''What are your most important activities in your job?'' Even if the latter question is followed by a list of possible activities, the question itself is essentially an open one. If response alternatives are incorporated in the question itself, the question should be coded as a closed question, for example: ''Do you work for yourself, in a family business or for someone else?''

The codes for ''answers'' partly parallel those for ''questions.'' Responding with ''yes'' or ''no'' to a yes/no questions is also coded 'Y', selecting a response alternative is coded 'A', and an ''open'' answered is coded 'O'. In addition a refusal to answer and ''don't know'' are coded as 'r' and 'b' (for ''blank''), respectively. If ''don't know'' is one of the scripted response alternatives, it should be coded 'A', however.

Perception (coded 'P' on EXCHANGE) of an utterance can be shown by repeating or summarising that utterance (code 'E', for ''Echo''), or by saying ''hm–mm,'' ''OK,'' etc. (coded 'n'). In addition we use code 's' to indicate a silence and 'f' to indicate a filled pause (like ''uhm'').

Requests can be requests for repetition ('d', for ''duplicate''), or requests to explain the meaning of a question or an answer ('m').

Further, comments can be either personal ('p') or task-oriented ('t'). Examples of task-oriented comments are ''Good question!'', ''That's useful information.'' Examples of personal comments are ''How nice for you,'' ''I understand how you feel.'' Especially personal comments add an affective factor to the coding scheme. Dijkstra (1987) and van der Zouwen, Dijkstra, and Smit (1991) have shown that interviewers, using such comments, definitely affect responses.

Finally, detours can be task-oriented ('t') or personal ('p') too. Examples of task-oriented detours are ''One moment please, I have to switch the tape of the tape recorder,'' or a discussion between interviewer and respondent about the sampling procedure. Examples of personal detours are ''What a nice picture on the wall,'' or remarks about the respondent's baby.

These four coding variables can be applied to many different kinds of interactions. For example, we applied these same variables to the introduction of an interview, to describe the process of persuading persons to participate in an interview. For example, a refusal by the respondent (''No, I don't want to participate'') is coded 'R' on ACTOR, 'A' on EXCHANGE, 'O' on DISTANCE and 'Y' on SPECIFICATION. A reason for refusing (for example ''I'm too busy'') is '1' on DISTANCE and 'O' on SPECIFICATION. These four coding variables can also be used for unstandardised or open interviews. For example, the interviewer can present (unscripted) answer alternatives in an open interview.

3.6. Adequacy

The fifth coding variable concerns the ADEQUACY of an utterance. This coding variable is specific to standardised survey interviews: most categories make sense only in the context of such an interview. If the interviewer reads the scripted question as worded, this utterance is coded adequate (A); if the respondent selects one of the response alternatives, this is marked adequate too; an utterance by an actor that is correctly repeated or summarised by the other actor is also coded adequate, and so on.

An utterance is coded ''invalid'' ('I') if the interviewer changes the meaning of a question; or if the respondent apparently misunderstood the meaning of the question. In example 1, the third utterance (''yes'') should be coded invalid, even if from the rest of the sequence it had emerged that someone did break into the respondent's home: at the moment of the answer this answer concerns a question with a meaning different from the one intended by the researcher.

Questions can also be posed in a suggestive manner ('S'), for example, if the interviewer suggests a particular response alternative to the respondent, not justified by previous answers of the respondent.

Finally, the category ''mismatch'' ('M') is used for a respondent's answer that does not match one of the response alternatives (for example, ''not many'' instead of ''few,'' where ''not many'' is not an alternative.) This code is also used if another actor's utterance is incorrectly repeated or summarised, or if a question deviates from the scripted version, yet without altering the meaning.

Example 2:

ACTOR	EXCHANGE	DISTANCE	SPECIFICATION	ADEQUACY	Verbal utterance	Description
I	Q	Ø	I	A	I: And now we have some questions about government agencies	introduction to question from questionnaire
I	Q	Ø	I	A	I: As you know, every ten years there is a census of the population of the United States	introduction to question from questionnaire
I	Q	Ø	C	A	I: How confident are you that the Census Bureau protects the privacy of personal information about individuals and does not share it with other government agencies?	I reads scripted question
I	Q	Ø	A	A	I: Very confident, somewhat confident, not too confident, or not at all confident?	I presents response alternatives
R	R	Ø	m	x	R: Share it with what other governments?	R requests explanation
I	Q	Ø	C	A	I: Well the question doesn't specify, it just says other government agencies	I just repeats (part of) question
R	A	Ø	A	M	R: Oh, probably very confident	R gives 'mismatch' answer
I	P	Ø	n	x	I: Okay	I signifies that answer is perceived

Categorising utterances on the adequacy variable does not always make sense. For example, utterances not having code 'O' on distance are not further categorised on adequacy. Similarly, a request for repetition can always be viewed as an adequate action. Distinguishing between adequate and inadequate requests does not make sense. In such cases the adequacy variable obtains code 'x'. Example 2 illustrates these five coding variables.

The first two utterances concern the scripted introduction ('I' on SPECIFICATION) to the question. Both utterances are read as worded and hence coded 'A' on ADEQUACY. Note that the introduction is made up of two utterances; both are separately coded. In this way it is possible to identify problems that occur with each one of these utterances, or if one of them is omitted. The third utterance is a closed question ('C'), followed by the response alternatives ('C', respectively 'A' on SPECIFICATION). Both utterances equal the scripted version and are coded adequate. Next the respondent asks for an explanation: it is a request ('R' on EXCHANGE) to explain ('m' on SPECIFICATION) a question from the questionnaire ('O' on DISTANCE). The interviewer does not give such an explanation, but merely repeats (part of) the question; this is coded exactly the same as posing the question itself. The utterance is coded adequate; the meaning of the question has not changed. Note that ''adequate'' does not necessarily mean that the action itself is appropriate. It may have been more appropriate to explain the meaning of the question. However, this is of no concern here, because the sequence of codes contains all relevant information (a request for explanation is followed by a repetition of the question). It is for the researcher to decide whether or not this part of the sequence should be regarded as appropriate. The coder only has to decide whether the repetition is adequate in the sense that the meaning is not changed or no answer is suggested.

The respondent answers ''probably very confident.'' This answer does not match one of the scripted response alternatives (''very confident, somewhat confident, not too confident, not at all confident''). ''Probably very confident'' is certainly less then ''very confident.'' Hence the utterance is coded mismatch ('M' on ADEQUACY). The interviewer should have repeated the alternatives, or at least part of them. The sequence is closed with a short statement signifying perception (''Okay'', coded 'n' on SPECIFICATION). The adequacy of such short statements is not ascertained, and coded 'x'.

3.7. Extending the coding scheme

Although these five coding variables can describe the interaction process in great detail to answer particular research questions, other coding variables can easily be added. For example, if a survey interview includes questions with response alternatives, adding information on the particular response alternative selected by the respondent may be useful. In this way any discrepancies between the respondent's answer and the score as is written down by the interviewer can easily be ascertained. Similarly, in the case of suggestive behaviour, it may be interesting to know which alternative in particular is suggested by the interviewer. We will call this coding variable the DIRECTION of an utterance.

Essentially, the code equals the number of the response alternative mentioned by the respondent, or suggested by the interviewer. If the interviewer reads all alternatives, this is coded 'a' If only a subset is read, code 's' is used. If the utterance refers to a show

Example 3:

ACTOR	EXCHANGE	DISTANCE	SPECIFICATION	ADEQUACY	DIRECTION	Verbal utterance	Description
I	Q	Ø	C	A	x	I: And sales taxes	I reads scripted question
R	A	Ø	A	A	2	R: eh-uh, the second one	R selects response alternative
I	P	Ø	E	A	2	I: About right	I signifies perception of answer
R	A	Ø	A	A	1	R: Oh no, excessively high	R selects response alternative
I	P	Ø	n	x	x	I: Okay	I signifies that answer is perceived

card (‘‘You can pick an alternative from this card’’), code ‘c’ applies. If the direction cannot be determined because of a ‘‘mismatch,’’ we will use codes like ‘l’ (‘low’) or ‘h’ (‘high’) to indicate the direction; in Example 2 above, ‘‘Probably very confident’’ is coded ‘1’ because the response ranks among the low-numbered alternatives (‘‘very confident’’ is 1, ‘‘somewhat confident’’ is 2, ‘‘not too confident’’ is 3, and ‘‘not confident at all’’ is 4). If the direction cannot be determined because the utterance is interrupted by another speaker (for example if the interviewer is interrupted when reading the response alternatives) this is coded ‘i.’ If the direction does not make sense (for example the interviewer reads the question in a non-suggestive manner) code ‘x’ is used.

Example 3 illustrates this coding variable. The Q–A sequence in this example was part of a number of questions on taxes. Preceding these questions, the response alternatives were read to the respondent by the interviewer: ‘‘For each type of tax, please tell me whether you think the tax is excessively high, about right, or low.’’ The question about sales taxes was the third one of this series.

The codes for DIRECTION do not need further explanation in this example. Note that the second utterance (‘‘The second one’’) is coded ‘‘adequate’’ on ADEQUACY and not as ‘‘mismatch,’’ because it unambiguously points to alternative number 2. Nor is it ‘‘invalid:’’ the respondent did not misunderstand the meaning of the question. The example is interesting because the coded sequence shows that the respondent gives two different answers. This would not be visible if DIRECTION had not been added to the coding variables. Giving two different answers to a question in one Q–A sequence may point to particular problems; here because the respondent apparently forgot the complete set of response alternatives, because they were read earlier.

3.8. Conclusions

According to this coding system, a verbal utterance is represented by a code string, consisting of the codes on all coding variables. For example, the code string ‘‘IQØAAa’’ means that the interviewer reads all alternatives as worded in the list. The sequence of code strings:

IQØNAx IQØAAa RAØAM1 IQØAS1 RAØAA1 IPØnxx

represents the verbal utterances of the **Q–A** sequence. Given the question and response alternatives, the course of actions can fairly well be reconstructed. According to the coded sequence, one can infer that the interviewer reads both question and alternatives as worded in the questionnaire, at least without significant alterations; that the respondent gives an answer which does not match one of the alternatives, but ranks among the low-numbered alternatives. The interviewer ''successfully'' suggests alternative number 1. The sequence is closed with a short statement (like ''Okay'') by the interviewer. Of course these reconstructions will be less precise if utterances concern for example ''detours'' or irrelevant utterances (distance 3).

Other advantages of this coding system are:

- The coding task is divided into a number of subtasks, each with a limited number of codes, which makes it easy to formulate decision rules describing differences between codes.
- A coder may take a wrong code on a particular coding variable; nevertheless, the codes on the other coding variables may still be valid.
- Although the number of different codes is small, the number of different code *combinations* is very large. This means that even infrequently occurring types of utterances can adequately be represented, without introducing new codes. For example, suppose the respondent asks the interviewer his or her opinion about a particular question, this will be coded as: RQOCAx.
- The system very well suits computer assisted coding. The next section will illustrate this.

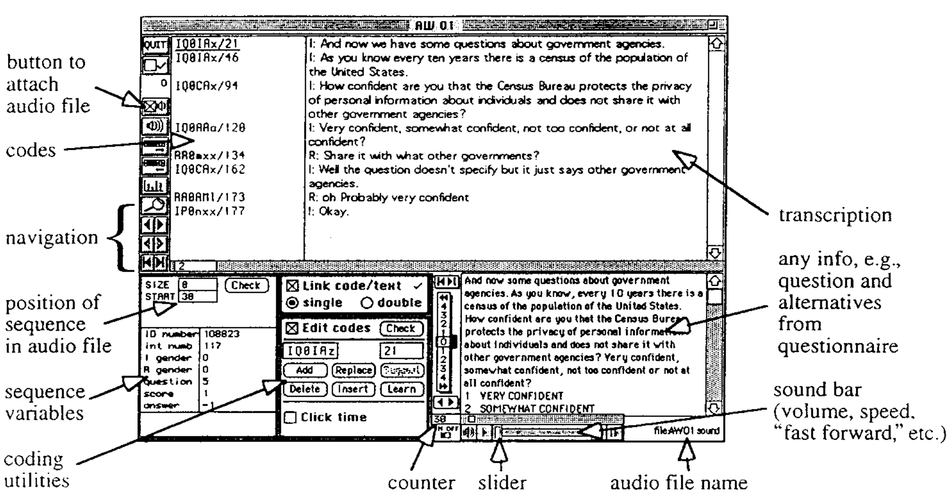

Fig. 1. A SEQUENCE VIEWER ''card.''

4. The Sequence Viewer Program

In computer assisted coding with the SEQUENCE VIEWER program, the following steps are taken (see Figure 1).

1. Record taped **Q–A** sequences as sound files and video-recorded **Q–A** sequences as movie files on hard disk. One second of sound of sufficient quality takes about ten kilo byte; a one-second movie takes about 100 kilo byte. Hence, devices to store large amounts of data are necessary. We store these files on CD-ROM afterwards; a CD-ROM can hold about 18 hours of mono-sound, sampled at 11 kHz.
2. Create a ''card'' for each **Q–A** sequence, in the SEQUENCE VIEWER program and connect the appropriate sound file to it. Each **Q–A** sequence appears on a separate card. Also, each card contains information on so-called ''sequence variables'' (see lower left corner of Figure 1). These variables describe characteristics of the sequence itself, for example question number, respondent sex, etc.
3. Transcribe the **Q–A** sequence on the card from the sound files on disk. A great advantage over transcribing from tape recordings is that the program allows immediate access to any part of the **Q–A** sequence stored as sound file. To speed up the transcription process, we usually put the scripted versions of questions and response alternatives in advance in the transcription field. For example, in the case of the question in Example 1 and Figure 1, the first four utterances (introduction, question proper, and alternatives) are in the transcription field already. The transcriptor only has to listen carefully to the spoken text of these four utterances, and to adjust the transcripts only if necessary. For example, to the first utterance the words ''we have'' should be added (as can be seen in Figure 2 by comparing the transcript and the scripted text in the lower right field). Of course, additional utterances should be completely transcribed.
4. Code the sequences. Because text and sound are linked it is very easy to take the pronunciation of utterances into account when deciding on the most appropriate code. Moreover, the program immediately checks on typing errors (for example non-existing ''codes''). To this end, the researcher can enter the codes that are allowed for a particular coding variable.

A great advantage for the coding process is that the spoken text is immediately available. Intonation, etc. is often essential in deciding on the meaning of utterance and hence on the correct code. For example, a rising voice can mean the difference between a repetition and a question. Whether or not a voice rises at the end of an utterance may be difficult to hear and can only be decided after playing the sound a number of times. In this way the coding process uses both the transcriptions (which makes it easy to detect for example deviations from scripted questions) and the spoken utterances; the coder is provided with optimal information.

Even more important in the coding process is that the program can suggest the most appropriate code string. For example, in a survey the question ''How confident are you that the Census Bureau protects the privacy of personal information about individuals and does not share it with other government agencies?'' from Example 2 may have been posed to a large number of respondents. Each time this question is posed as worded, it should obtain the code string 'IQØCAx.' Similarly, the introduction to this question, and the response alternatives are coded IQOIAx and IQOAAa, if they are read as worded. A response like ''excessively high'' will likely be coded as RAOAA1. And so on. The program allows the researcher to enter such associations, with the effect that the program suggests the associated code string each time this utterance should be coded. In addition, the computer produces the agreement between the utterance to be coded and the utterance as entered by the researcher. If a question is not read as worded, for example, the program will suggest the associated code string, and in addition will show the degree of agreement with the scripted version; e.g., '87%.'[3] In such a case it is of course for the coder to decide whether a code like 'I' (for ''invalid'' on the coding variable ADEQUACY) is more appropriate than 'A.' Moreover, the system is adaptive and can be adjusted all along the coding process. For example, the interviewer in the three examples above appeared to end nearly all sequences with ''Okay,'' to be coded as IPOnxx. The coder can temporarily add such an association.

The suggestion option not only speeds up considerably the tedious coding work but also improves the reliability of the codes. However, a high reliability will also be obtained if coders uncritically accept the suggestions from the program, also if a suggestion is wrong. Coders should be trained to carefully use the suggestion option and not rely blindly on the suggestions. To investigate the reliability of the coding, minimising this effect, two interviews[4] were coded twice, once by an experienced coder using the suggestion option and once by the author, not using this option of course. Taking all six coding variables together, the reliability (Cohen's kappa), appeared as 0.78 and 0.80, respectively. It should be noted that these reliabilities are underestimates in a sense. If two code strings differed with respect to only one coding variable, e.g., IQOAAs and IQOAAi, they were conceived as unequal, irrespective of the fact that both coders agreed that the utterance concerned the interviewer adequately reading part of the response alternatives from the questionnaire; they only differed when it came to whether reading only a part of the alternatives was caused by an interruption by the respondent or not.

Codes can also be provided with time information. Figure 1 shows the offset times after the codes (after the slashes). Assigning the offset time to a code is simply accomplished by clicking the mouse on a particular code, as soon as the utterance belonging to this code is finished, as can be heard if one plays the sound. Not only can the addition of time information be valuable for further analyses, it is also used by the program for immediate access to each utterance from the sound file. For example, if one clicks on the transcripted utterance 'R': Share it with what other governments?'' (Figure 1), immediately the corresponding utterance is played from the sound file. Finally, the codes described in Section 3 are not inherent to the SEQUENCE VIEWER program. The researcher can define his or her own code system for a particular datafile. All information about the code system used, for example the allowed codes, is stored together with the data themselves.

[3] For this purpose, the program uses a complex algorithm to calculate the extent to which an utterance resembles another utterance.
[4] The interviews came from a telephone survey after watching commercials on television. From each of the two interviews 37 questions were analysed. The total number of coded utterances is 422.

5. The Analysis of Interviewer-respondent Interaction: The Sequence Program

Many studies of interviewer-respondent interaction in the survey interview analyse data by counting the number of times particular codes occur. Such analyses can be characterised as frequency analysis rather than sequence analysis. However, the occurrence of inadequate interviewer and respondent behaviour cannot be merely ascribed to bad interviewers or bad questions. The eventual response to a survey question can be viewed as the result of the interaction between interviewer and respondent. For example, a ''mismatch'' answer of the respondent (an answer that does not match one of the prescribed response alternatives) may cause the interviewer to suggest a response alternative, which in turn may yield an adequate answer by the respondent (if the suggestion is affirmed). Moreover, in such a case the latter response is counted in a frequency analysis as ''adequate,'' thus neglecting the fact that this response may very well not be valid, or at least not obtained properly.

Michael Brenner (1982, 1985) was among the first researchers who explicitly paid attention to the sequence of verbal behaviour involved in answering a survey question. Although he developed a computer program that was able to calculate frequency distributions of actions, the most interesting analyses, tree representations of interviewer-respondent interactions, were calculated by human beings. A tree representation starts from a particular action and shows in how many cases this action is followed by a different action. In turn each of these actions shows how many times it is followed by other actions; and so on (see Figure 2 for an illustration of such a tree).

Another interesting approach to the study of sequences of interviewer and respondent behaviours is identifying straightforward or unproblematic sequences; usually called paradigmatic sequences (Schaeffer and Maynard 1996). Sykes and Morton-Williams (1987) give two examples of such sequences (p. 201):

1. Question asked as worded
↓
Answer given adequate for coding
↓
Response acknowledged by interviewer

and

2. Question asked as worded
↓
Answer given adequate for coding
↓
Interviewer confirms response given
↓
Respondent repeats answer
↓
Response acknowledged by interviewer

They calculated the percentage of straightforward sequences for each survey question, with a low percentage indicating potential problems with that question. In 1992 Sykes and Collins elaborated on this concept. Their article describes in great detail how a complex recoding process of the coded utterances was applied to identify such straightforward sequences. Further recoding enabled them to classify a number of types of non-straightforward sequences and to relate these types to question types (for example questions using showcards and multiple response questions). Such analyses are quite intriguing, but yet do not answer the question why things go right or wrong because of particular events in the interaction itself. More generally, one might ask questions like ''Which respondent behaviour causes suggestive interviewer behaviour?''; ''How do interviewers handle problems like requests for explanation or 'mismatch' answers?''; ''What kinds of non-paradigmatic sequences can be distinguished?'' We developed the SEQUENCE program (Dijkstra 1994) to be able to answer such questions.

The SEQUENCE program uses files that consist of sequences of code strings (like the example in Section 3.8). In addition, each sequence is preceded by the values of the sequence variables belonging to the sequence, representing for example interviewer and question number, sex of interviewer and respondent, and so on. Such files can be created by the SEQUENCE VIEWER program. Existing ASCII files with appropriate data can usually be made suitable by some simple replace operations. The SEQUENCE program allows us to perform a large number of analyses. In addition to usual analyses like lag-sequential analysis it is especially suited for studying characteristics of sequences as a whole, for example by relating the occurrence of particular patterns in sequences, or the degree of agreement with a paradigmatic sequence to sequence variables. Questions like ''Do interviewers differ with respect to the proportion of paradigmatic sequences they produce?'' are easily and quickly answered.

A great deal of attention has been given to making the program more user friendly. A set-up for simple analyses can be made in seconds by only a few mouse clicks. Most analyses are performed very fast; for example the tree to be discussed in Section 5.2 was created in less than a second. Numerous options are available for finding and replacing particular codes.

5.1. Overview of analyses in SEQUENCE

In this section we will give a short (not exhaustive) overview of the analyses that are possible with SEQUENCE.

Frequencies: Counts the number of times that different codes occur. Frequencies can be tabulated against the values of a particular sequence variable, for example to compare the frequency distribution of codes for different interviewers.

Next event and *previous event*: Counts how often different codes follow (or precede) a given code, with a lag between 1 and 9, including expected frequencies.

Matrix: Calculates a transition matrix of codes, with a lag between 1 and 9. Also calculates z-values, expressing whether transitions occur above or below chance and general measures of dependencies.

Tree: Shows sequences in the form of a tree structure, given a particular coded utterance to start with. In Section 5.2 this analysis is discussed in more detail.

Path: Shows sequences in the form of a number of different paths, given a particular code to start with, and a particular code to end with. That is, this analysis answers questions like ''What is happening between a request for explanation and an improper (''mismatch'') answer?''

Patterns: Finds complex patterns of codes in a sequence; for example ''find all sequences containing a 'mismatch' response by the respondent, *not* followed somewhere in the sequence by a correct presentation of the response alternatives by the interviewer, but instead by 'summarising' the response by a particular alternative from the list.'' A new sequence variable can be created with (for example) the value '1' if this pattern is present in a sequence, and with '0' if this pattern is not present; for example to investigate whether the occurrence of particular errors is related to the interviewer (number).

Count sequences: Frequency distribution of sequences (the number of times a particular sequence occurs).

Compare sequences: Calculates the degree of agreement between two sequences as a number between 0 (completely different) and 1 (exactly equal).

Agreement matrix: Calculates a matrix of agreements between sequences.

Cluster sequences: Performs a cluster analysis on sequences.

Reliability: Performs a reliability analysis (using kappa as a measure) on utterances that are coded twice.

5.2. Example of an analysis with SEQUENCE

An interesting research question is what happens after the respondent gives a ''mismatch'' answer (an answer not fitting one of the scripted response alternatives) to a question from the questionnaire. Such a research question can easily be answered by, for example, the ''tree'' procedure. The data for this example came from a face-to-face survey among young adults, who were asked large numbers of questions about education, work, and partner relationships. We coded 1227 **Q–A** sequences concerning 9 questions and 17 interviewers.

Figure 2 shows the set-up screen for this analysis. The code string for a mismatch answer is RAOAM. The datafile also contained a sixth coding variable, equal to the DIRECTION variable discussed in Section 3.7. However, for the present research question this coding variable is not of interest and so can be neglected. This is indicated with the full stop (·) on the sixth position of the code string that appears after the ''Root event,'' i.e., the code string of the utterance we want to start with. In order to prevent the tree becoming too large, we limit ourselves to the first four coded utterances after an inadequate answer ('Tree depth = 4 + 1 = 5'). Moreover we neglect transition frequencies between utterances that are less than 3: transitions that occur only once or twice do not give much insight into patterns or regularities, and will make the tree too large.

From the set-up it can also be seen that ''one branch per sequence'' is checked. This

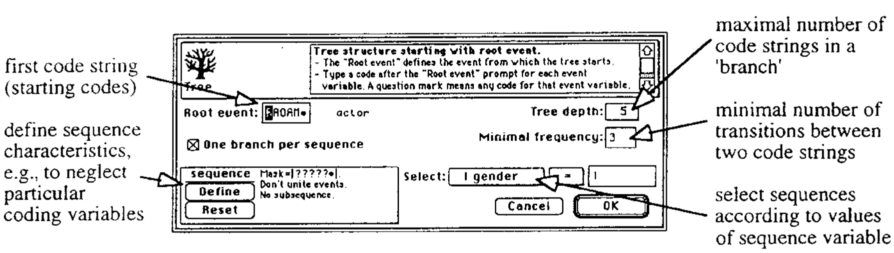

Fig. 2. Set-up for tree analysis.

Table 1. Output of tree analysis

1	→ 261 RAØAM·				
2		→ 10 IPØnx·			
3		→ 27 IPØEA·			
4			→ 5 RAØAA·		
5			→ 6 RAØAM·		
6		→ 18 IPØEM·			
7			→ 8 RAØAA·		
8		→ 103 IQØAA·			
9			→ 82 RAØAA·	→ 13 IPØnx·	
10				→ 26 IPØEA·	
11				→ 7 IQØAA·	
12					→ 7 RAØAA·
13			→ 11 RAØAM·		
14				→ 4 IQØAS·	
15		→ 5 IQØAM·			
16			→ 4 RAØAM·		
17		→ 26 IQØAS·			
18			→ 19 RAØAA·		
19				→ 3 IPØnx·	
20				→ 3 IPØEA·	
21			→ 4 RAØAM·		
22		→ 5 IQØEA·			
23		→ 3 IQØEM·			
24		→ 23 RA10A·			
25			→ 3 IP1nx·		
26			→ 3 IQØAA·		
27			→ 3 IQØAS·		

means that if a respondent utters two or more ''mismatch'' answers, only the first one serves as a starting point for the tree. In this way dependencies in the data are prevented (the same sequence can attribute only once to the tree).

One can create a tree for a particular value on a coding variable. For illustrative purposes, in the set-up in Figure 2 only sequences from male interviewers ('I sex = 1') are selected. However, we actually performed the analysis on all 1227 sequences.

Table 1 shows the output of the tree analysis and reads as follows. Line 1 (line numbers do not show up in the actual output) shows that 261 sequences contain a ''mismatch'' answer by the respondent (code RAØAM·). These 261 mismatches are followed in 10 cases by the interviewer signifying perception (line 2; IPØnx·), in 27 cases by the interviewer correctly repeating the (mismatch) answer (line 3; IPØEA·), and so on. These numbers (10, 27, etc.) do not add up to a total of 261 because transitions less than 3 are not included.

Out of these 261 sequences containing a mismatch, this answer is followed in 103 sequences by the interviewer's adequately presenting the response categories (line 8; IQØAA·). In 82 out of these 103 sequences, the respondent selects an adequate answer (line 9; RAØAA·), but in 11 cases the respondent persists in inadequate answering (line 13; RAØAM·).

It shows that an inadequate answer of the respondent is followed by a suggestion by the interviewer in 26 sequences (line 17; IQØAS·), which in turn yields an adequate answer by the respondent in 19 sequences (line 18; RAØAA·), that is, a selection from the set of response alternatives. Needless to say, these answers should not be trusted.

6. Conclusions

In the introduction three problems were mentioned in studying interviewer–respondent interactions in survey interviews: 1) how can the sequences of actions of interviewer and respondent be represented; 2) how can these actions efficiently and reliably be coded; 3) and how can the coded sequences be analysed. The procedures we developed are particularly suited to studying complex dependencies between actions of interviewer and respondent, and to detecting patterns or regularities in these dependencies. Because of the diverse nature of interviewer–respondent interactions, we devised a coding scheme that enables us to choose among this wealth of different actions. Moreover, for detecting regularities in the interaction process, it was necessary to develop procedures for transcribing and coding vast numbers of **Q–A** sequences.

By now we have transcribed and coded many thousands of **Q–A** sequences, and it appeared that this could be done in a fast and reliable way. Coding and transcribing an interview takes at least twice the time of the interview itself (the sound should be played twice). Actually, it takes even longer. The actual speed depends on the extent that the **Q–A** sequence deviates from paradigmatic sequences. Transcribing and coding sequences that are more or less paradigmatic proves to be extremely fast (and reliable). Long and non-paradigmatic sequences (we have found sequences with over 200 utterances) take much more time. Most utterances should be typed in and the suggestion option for quick coding is of less use, because such sequences tend to deviate completely from the scripted question. Transcribing and coding interviews with many nonparadigmatic sequences may take up ten times the duration of the interview itself.

Analysing these large datasets has only just begun. A first example can be found in Van der Zouwen and Dijkstra (1995), where we were able to characterise sequences according to four different types and relate these types to their causes and to their effects on the obtained answers.

7. References

Barath, A. and Cannell, C.F. (1976). Effect of Interviewer's Voice Intonation. Public Opinion Quarterly, 40, 370–373.

Belson, W.A. (1967). Tape Recording: Its Effect on Accuracy of Response in Survey Interviews. Journal of Marketing Research, 4, 253–260.

Belson, W.A. (1969). The Semantic Differential Scaling System in Market Research III – Interviewer Deviation from Instructions. Survey Research Centre, London School of Economics and Political Science.

Blair, E.A. (1978). Nonprogrammed Speech Behaviors in a Household Survey (unpublished doctoral dissertation). University of Illinois: Department of Business Administration.

Brenner, M. (1982). Response-Effects of ''Role-Restricted'' Characteristics of the

Interviewer. In W. Dijkstra and J. van der Zouwen (eds.): Response Behaviour in the Survey-Interview. London: Academic Press, 131–165.

Brenner, M. (1985). Survey Interviewing. In M. Brenner, J. Brown, and D. Canter (eds.): The Research Interview; Uses and Approaches. London: Academic Press, 9–36.

Cannell, C.F., Lawson, S.A., and Hausser, D.L. (1975). A Technique for Evaluating Interviewer Performance. Ann Arbor: The University of Michigan.

Cannell, C.F. and Oksenberg, L. (1988). Observation of Behavior in Telephone Interviews. In R. Groves, P. Biemer, L. Lyberg, J. Massey, W. Nicholls II, and J. Waksberg (eds.): Telephone Survey Methodology. New York: Wiley, 475–495.

Chapple, E.D. (1949). The Interaction Chronograph; Its Evolution and Present Application. Personnel, 25, 295–307.

Dijkstra, W. (1987). Interviewing Style and Respondent Behavior: An Experimental Study of the Survey-Interview. Sociological Methods and Research, 16, 309–334.

Dijkstra, W. (1994). SEQUENCE, A Program for Analysing Sequential Data. Bulletin de Méthodologie Sociologique, 43, 134–142.

Ericsson, K.A. and Simon, H.A. (1993). Protocol Analysis: Verbal Reports as Data. Cambridge, MA: MIT Press.

Fowler, F.J. and Cannell, C.F. (1995). Using Behavioral Coding to Identify Cognitive Problems with Survey Questions. In N. Schwarz and S. Sudman (eds.), (1996): Answering Questions. Methodology for Determining Cognitive and Communicative Processes in Survey Research. San Francisco: Jossey-Bass, 15–36.

Hartmann, P. (1992). Interviewing when the Spouse is Present. Paper presented at the International Conference on Social Science Methodology, Trento.

Koomen, W. and Dijkstra, W. (1975). Effects of Question Length on Verbal Behavior in a Bias-reduced Interview Situation. European Journal of Social Psychology, 5, 399–403.

Marquis, K.H., Cannell, C.F., and Laurent, A. (1972). Reporting of Health Events in Household Interviews: Effects of Reinforcement, Question Length and Re-interviews. Vital and Health Statistics, Series 2, No. 45.

Matarazzo, J.D., Saslow, G., and Matarazzo, R.G. (1956). The Interaction Chronograph as an Instrument for Objective Measurement of Interaction Patterns during Interviews. Journal of Psychology, 41, 347–367.

Matarazzo, J.D. and Wiens, A.N. (1972). The Interview. Research on Its Anatomy and Structure. Chicago: Alaine Atherton.

Morton-Williams, J. (1979). The Use of ''Verbal Interaction Coding'' for Evaluating a Questionnaire. Quality and Quantity, 13, 59–75.

Oksenberg, L., Cannell, C., and Kalton, G. (1991). New Strategies in Pretesting Survey Questions. Journal of Official Statistics, 7, 349–365.

Prüfer, P. and Rexroth, M. (1986). On the Use of the Interaction Coding Technique. Mannheim: ZUMA-Arbeitsbericht Nr. 86-05.

Schaeffer, N.C. and Maynard, D.W. (1996). From Paradigm to Prototype and Back Again: Interactive Aspects of Cognitive Processing in Standardized Survey Interviews. In N. Schwarz and S. Sudman (eds.) Answering Questions. Methodology for Determining Cognitive and Communicative Processes in Survey Research. San Francisco: Jossey-Bass, 65–88.

Sudman, S., Bradburn, N.M., and Schwarz, N. (1996). Thinking about Answers: The Application of Cognitive Processes to Survey Methodology. San Francisco: Jossey-Bass.

Sykes, W. and Collins, M. (1992). Anatomy of the Survey Interview. Journal of Official Statistics, 8, 277–291.

Sykes, W. and Morton-Williams, J. (1987). Evaluating Survey Questions. Journal of Official Statistics, 3, 191–207.

Tucker, C. (1997). Methodological Issues Surrounding the Application of Cognitive Psychology in Survey Research. Bulletin de Methodologie Sociologique, 55, 67–92.

van der Zouwen, J. and Dijkstra, W. (1995). Trivial and Non-trivial Question–Answer Sequences; Types, Determinants and Effects on Data Quality. Proceedings of the International Conference on Survey Measurement and Process Quality, Bristol, April 1995, 81–86.

van der Zouwen, J., Dijkstra, W. and Smit, J.H. (1991). Studying Respondent-Interviewer Interaction: The Relationship Between Interviewing Style, Interviewer Behavior, and Response Behavior. In P.P. Biemer, R.M. Groves, L.E. Lyberg, N.A. Mathiowetz, and S. Sudman (eds.): Measurement Errors in Surveys. New York: Wiley, 419–437.

Wiens, A.N., Matarazzo, J.D., and Saslow, G. (1965). The Interaction Recorder: An Electronic Punched Paper Tape Unit for Recording Speech Behavior During Interviews. Journal of Clinical Psychology, 21, 142–145.

Wilcox, K. (1963). Comparison of three Methods for the Collection of Morbidity Data by Household Surveys. P. D. dissertation, The University of Michigan, School of Public Health.

Received August 1995
Revised November 1997