Codebook notes

Article: Developing and Using a Codebook for the Analysis of Interview Data: An Example from a Professional Development Research Project

Codes, Codebooks, and Coding

Codes are defined as ‘‘tags or labels for assigning units of meaning to the descriptive or inferential information compiled during a study’’ (Miles and Huberman 1994: 56), and their development1 is the initial step in analyzing interview data. To ensure meaningful labels, codes are assigned to chunks of data, usually phrases, sentences, or paragraphs that are connected to a specific context or setting (Miles and Huberman 1994).

The actual process of coding is an integral part of the interview data analysis process. Coding is the assigning of codes (that have been previously defined or operationalized in a codebook) to raw data

According to Corbin and Strauss (2008), there are two major levels of coding—open coding and axial coding. When beginning to code interview data, the first step is to engage in the process of open coding or ‘‘breaking data apart and delineating concepts to stand for blocks of raw data’’ (Corbin and Strauss 2008:195). Open coding allows for exploration of the ideas and 138 Field Methods 23(2) meaning that are contained in raw data. While engaging in open coding, the researcher creates codes or concepts. Once codes have been created using open coding, it is necessary to analyze them through the process of axial coding. This higher level of coding enables researchers to identify any connections that may exist between codes. When beginning the analysis process, inexperienced qualitative researchers are likely to have many questions, including the central question: ‘‘How do I create a codebook?’’ However, another question they should ask is, ‘‘What role does theory play in the creation of a codebook?’’ Similarly, once a codebook has been created, they may discover they need to ask, ‘‘How do I train others to use a codebook?’’ Questions such as these can frustrate and stymie the efforts of beginning researchers. Therefore, for the remainder of this article, we respond to these questions and describe how we created a codebook for analyzing interview data as part of our multiyear funded research project.

Creating a Codebook

As previously mentioned, codes are created from three major areas including theory (theory-driven), data (data-driven), and research goals (structural). In the case of NMD, only theory- and data-driven codes were created to assist in the coding of interviews. Boyatzis (1998) indicates that there are separate procedures for creating theory- and data-driven codes. Developing theory-driven codes involve three steps: (1) generate the code; (2) review and revise the code in context of the data; and (3) determine the reliability of coders and the code. Data-driven codes, on the other hand, involve five steps to inductively create codes for a codebook: (1) reduce raw information; (2) identify subsample themes; (3) compare themes across subsamples; (4) create codes; and (5) determine reliability of codes. We will use Boyatzis’s framework to demonstrate the steps we used to create theory- and data-driven codes and codebook definitions. (See Figure 2 for a visual of the steps for creating a codebook.)

How Do You Develop Theory-Driven Codes?

The first step in developing theory-driven codes is to create codes. Codes are generated from the theories that guide the research. The second step in developing theory-driven codes is reviewing and revising the codes in context. This necessitated discussing the appropriateness of the code labels and how they were to be applied to the data. Our goal was to create code labels, as suggested by Boyatzis (1998), which were conceptually meaningful, clear and concise, and close to the data.

How Do You Develop Data-Driven Codes?

The first step in developing data-driven codes is to determine how to reduce raw information into smaller units, such as categories or themes. We discussed the possibility of coding line by line, on the sentence level, on the paragraph level, or by what we labeled the ‘‘level of meaning.’’ After reading several interviews, we realized that coding line by line and on the sentence level were often not meaningful. The paragraph level, on the other hand, often featured a variety of themes, making it impossible to label with only one code. Based on this, we decided to focus on the level of meaning. From this perspective, the ‘‘lumping’’ and ‘‘splitting’’ of text could occur at different locations, enabling a code to be made up of a line, sentence, or paragraph, as long as the essence is the same (MacQueen et al. 2008). The final step used in developing data-driven codes was to determine the utility/reliability of the codes using them to begin the analysis process.