CHAPTER

Analyzing qualitative data 11

11.1 INTRODUCTION

In Chapter 4, we discussed how to use significance tests to study quantitative data and measures such as speed, error rate, distance, adoption rate, and rankings. The well-defined nature of quantitative measures makes them appealing options for many studies. When we can clearly specify our measures of interest and how they are to be measured, research methods and analytic procedures can be clearly defined, making study design reasonably straightforward. Of course, complications may arise, but we don't need to worry about the definition of the underlying units used to measure task completion times.

Our discussions of case studies, interviews, and ethnography introduce markedly different kinds of data associated with research questions and analysis methods that are not quite so clear-cut. Rather than searching for numerical measurements, these qualitative studies attempt to study texts, observations, video, and artifacts to understand complex situations. Analysis of these data often raises challenges that rarely raise with quantitative data, as we struggle to interpret ambiguous comments and understand complex situations. To make matters worse, we don't even know what the "truth" is—as multiple researchers might (and often do) have different perspectives on the same situation.

Acknowledging these challenges, social science researchers have developed research methods designed to increase rigor and validity in analyzing qualitative data. Qualitative methods do not aim to eliminate subjectivity—instead, they accept that subjectivity is inherent to process of interpreting qualitative data, and they strive to show that interpretations are developed methodically to be consistent with all available data, and representative of multiple perspectives.

In this chapter, we present an introduction to qualitative research, discussing techniques for ensuring high-quality analysis of qualitative data that is both *reliable* and *valid.* We introduce the process of *coding,* which assigns labels to observations from text or other qualitative data forms. We specifically focus on grounded theory (Glaser and Strauss, 1967), the starting point for many qualitative analyses. The use of content analysis to extract categories from diverse "texts" is described, along with a discussion of the analysis of two very important forms of qualitative data: text and multimedia. In order to control the impact of subjective interpretation, a commonly accepted coding procedure should be adopted and

Research Methods in Human-Computer Interaction. http://dx.doi.org/10.1016/B978-0-12-805390-4.00011-X
© 2017 Elsevier Inc. All rights reserved.

statistical methods used to evaluate the validity and reliability of the coding completed by human coders. The general strategy discussed in this chapter is just one of the many approaches available for analyzing text and other qualitative information. Substantially different strategies may be used for different disciplines, such as literature or art.

11.2 GOALS AND STAGES OF QUALITATIVE ANALYSIS

The goal of qualitative analysis is to turn the unstructured data found in texts and other artifacts into a detailed description about the important aspects of the situation or problem under consideration. This description can take many forms, including textual narratives, graphical diagrams, and summary tables. These items can often be combined to provide the range of perspectives needed for understanding the underlying complexity.

According to Corbin and Strauss (Corbin and Strauss, 2014), qualitative data analysis consists of three stages. We start with a data set containing information about our problem of interest. For example, the problem can be related to challenges faced by or unique needs of a specific group of users (i.e., people with visual disabilities, senior citizens, children, etc.). It can focus on a specific technology such as a gesture-based input method, a photo sharing application, or 3D printing. The problem can also examine the interaction behavior in a specific context, such as text entry on a small screen while the user is constantly walking. Via analysis, we hope to identify major themes and ideas that describe the context, activities, and other perspectives that define the problem. In the second stage, we drill down into each component to find relevant descriptive properties and dimensions. In many cases, we need to understand not only the nature of each component, but also how they relate to each other. In the third stage, we use the knowledge we gained from studying each individual component to better understand the original substance and make inferences about that substance.

For example, we might analyze chat logs to study the online behavior of Internet users. Reading these logs, we might notice that three factors, namely personality, education, and computer-related experience, are repeatedly found to influence users' online behavior. We continue to study each of those three factors and how they relate to each other. We study the literature in psychology and sociology to understand the types of personality, how an individual forms and develops a specific personality, and how a specific type of personality affects an individual's social behavior. Once we have a thorough understanding of the three factors, we can tie the knowledge back to the original texts and examine how each of the components affects a user's online behavior. Specifically, we might use our literature review to identify specific personality or educational behaviors that might influence online behaviors, This application of experience and contextual knowledge is critical for the appropriate interpretation of qualitative data and the entire knowledge discovery process.

11.3 CONTENT ANALYSIS

Widely used in vastly different domains, *content analysis* refers to the process of developing a representative description of text or other unstructured input. Stemler (2001) summarized previous work (i.e., Berelson, 1952) and stated that content analysis is a systematic, replicable technique for compressing many words of text into fewer content categories based on explicit rules of coding. A broader definition proposed by Holsti (1969) allows for other types of information, defining content analysis as "any technique for making inferences by objectively and systematically identifying specified characteristics of messages" (p. 14). According to this definition, content analysis not only applies to textual information, but also to multimedia materials, such as drawings, music, and videos.

Content analysis is normally in-depth analysis that searches for theoretical interpretations that may generate new knowledge. As described by Corbin and Strauss (Corbin and Strauss, 2014), this type of analysis "presents description that embodies well-constructed themes/categories, development of context, and explanations of process or change over time" (p. 51). Although many people think content analysis is a qualitative research method, both quantitative and qualitative techniques can be used in the process of content analysis (Neuendorf, 2002).

11.3.1 WHAT IS CONTENT?

The target of content analysis usually covers two categories: media content and audience content. Media content can be any material in printed publications (e.g., books, journals, magazines, newspapers, and brochures), broadcast programs (e.g., TV or radio programs), websites (e.g., news websites, web portals, personal websites, or blogs), or any other types of recording (e.g., photos, films, or music).

Audience content is feedback directly or indirectly collected from an audience group. Audience content can be collected through a variety of methods such as surveys, questionnaires, interviews, focus groups, diaries, and observations. Traditionally, information collected via those methods is text based. In the HCI field, researchers and practitioners frequently collect both text-based information and multimedia-based information from the participants. Text or multimedia information used for content analysis can be collected through a variety of methods listed in Table 11.1. For more detailed information on each of those data collection methods, please refer to Chapters 5–9.

11.3.2 QUESTIONS TO CONSIDER BEFORE CONTENT ANALYSIS

Before you start analyzing the data, you need to consider several questions that can help frame the scope of the content analysis as well as the specific techniques that should be used for the analysis (Krippendorff, 1980).

First, you need to have a clear definition for the data set that is going to be analyzed. In some studies the definition of the data set is very straightforward. For example, if

Table 11.1 Major Categories of Content

Category	Subcategory	Examples
Media content	Publications	Books, journals, newspapers, brochures
	Broadcasting	TV programs, radio programs
	Websites	News, web portals, organizational websites, blogs
	Others	Films, music, photos
Audience content	Text	Notes from interviews, focus groups, or observations or diaries or surveys, text posts on social media
	Multimedia	Video- or audio-recording of interviews, focus groups, observations, or user studies, pictures or video recordings posted on social media

you interview 10 mobile device users on their daily usage of the device, your interview notes would be the data set that you are going to analyze. In other cases, the definition of the data set may not be that straightforward and special consideration is needed to select the appropriate content or messages that should be included in the data set. Suppose you want to study the development of interpersonal trust among members of an online community. The public messages that the community members leave on the bulletin board may contain valuable information. The messages that the community members exchange privately through applications such as Instant Messaging or email would also be useful. In this case, you need to consider the scope of your data set: Do you want to study the public messages only, the private messages only, or both? The answer to this question depends on both your research question and the practical issues of your study. If your research question is focused on the impact of the general community atmosphere and the sense of community on trust development, you may want to limit the data set to the public messages because they are the most relevant to your research question. In some cases, you may have to stick to the public messages because you have no access to the messages that are exchanged privately among the community members. You may also restrict the data set to messages posted during a specific time period. Overall, the scope of your data set may affect the key words or categories that you are going to use during the content analysis.

Once a clear definition of the data set is specified, you should study the data closely and remove any data that do not meet the criteria of the definition. In the online community study example, if you decide to study the public messages posted in 2015, then all of the private messages and any public messages from earlier or later years must be abandoned. If those messages remain in the data set and are analyzed, the data set is polluted and the results may be biased or misleading.

Content analysis studies should also clearly define the population from which the data set is drawn. This seems to be straightforward but many issues may be encountered

in practice. In the online community study example, the term "community members" may raise some questions. How do you define community members? Do community members include all the people who have visited the online community website? If the answer to this question is "yes," it means you are interested in examining not only those visitors who have posted messages, but also the people who have visited the website but have never posted a message (lurkers). If your data set only consists of the public messages, then the data comes from a subset of your population (those who have contributed by posting messages) and, therefore, it is not representative of the overall population. In this case, it may be more appropriate to restrict the target population of your study to those people who have posted messages. Examination of the data might reveal a wide range in the number of messages posted by participants. Some people might visit and post messages on a daily basis. Some people might only post one or two messages through the entire year. Do you count those extremely infrequent visitors as community members? If so, there may be concerns over whether the small number of postings from those visitors' limits the accuracy of the depiction of their opinions or behavior. Other factors that should be considered when defining the population include, but are not limited to, age, gender, profession, education, and domain experience.

Thirdly, you need to know the specific context of the data. Data analysis out of context is meaningless and highly biased. Any words, terms, and claims need to be interpreted in the specific context from which they are extracted. Consideration of the context is an iterative process, occurring at multiple levels throughout analysis. Before data analysis, you need to have a clear understanding of the higher-level context of your data set. For example, if you are studying the end-user's attitude toward security procedures in the organizational environment, you need to be aware that the type of business or profession may have a notable impact on the topic. An employee of a government agency who has access to classified information works in a very different environment from a staff of an entertainment facility. The government worker may have to go through security training on a regular basis while the staff working in the entertainment industry may have no security-related training. Therefore, the specific context of their work has great impact on the data that they provide. If you analyze their input without considering the context, it is like comparing apples with pears and the results are tainted. During the data analysis process, you need to consider lower-level context, such as the phrase, sentence, or paragraph. We discuss the interpretation of low-level context in Section 11.4.2.

11.4 ANALYZING TEXT CONTENT

11.4.1 CODING SCHEMES

Analyzing text content involves assigning categories and descriptors to blocks of text, a process called "coding." A common misunderstanding is that coding is nothing more than paraphrasing the text and counting the number of key words in the text. Actually, coding is much more than paraphrasing and key word counts. As stated by Corbin and Strauss (Corbin and Strauss, 2014), coding "involves interacting with

data, making comparisons between data, and so on, and in doing so, deriving concepts to stand for those data, then developing those concepts in terms of their properties and dimensions." A set of well-developed procedures for analyzing text content has been widely accepted in the social sciences and related fields. Because qualitative research is more vulnerable to bias than quantitative research, it is particularly important to follow the standard procedure to ensure the quality of the analysis and the robustness of the results.

Solid qualitative analysis depends on accurately identified concepts that later serve as "categories for which data are sought and in which data are grouped" (Blumer, 1969). The concepts and categories are also a means of establishing relations (e.g., correlation, causal relationships, etc.) between different entities. Identifying the coding categories can be a very daunting task for inexperienced researchers. The coding categories may come from several sources: an existing theoretical framework, the researcher's interpretation (research-denoted concepts), and original terms provided by the participants (in vivo codes).

There are two different approaches to analyzing the data: *emergent* coding and *a priori* coding. *Emergent* coding refers to the qualitative analyses conducted without any theory or model that might guide your analysis—you simply start by noting interesting concepts or ideas and continually refine those ideas until you are able to form a coherent model that captures the important details. *A priori* coding involves the use of an established theory or hypothesis to guide the selection of coding categories. These categories might come from previously published work in related areas, or from your own prior investigations of the topic at hand.

To illustrate the difference between these coding approaches, consider the earlier example of the study of online communities. Some studies might be based on the theory participants in the communities adopted various roles that defined the manner and content of their posts. These studies would use a priori coding, with codes selected to identify roles and their application. Other studies might be interested in understanding conversational dynamics more broadly, without any particular starting point. These studies would use emergent codes. Some studies might use a mixture of both methods.

The choice between emergent and a priori coding is often not straightforward. Existing theories and codes have the advantage of being somewhat simpler to use, at the potential costs of broader insight that might come from the more open-ended analysis associated with open coding.

11.4.1.1 Grounded theory and emergent coding

If you are working on a new topic that has very limited literature to build on, you may not be able to find established theories that allow you to develop the coding categories in advance. In this case, the emergent coding approach, based on the notion of *grounded theory,* is appropriate. Grounded theory was first proposed by Glaser and Strauss (Glaser and Strauss, 1967), who described a qualitative research method that seeks to develop theory that is "grounded in data systematically gathered and analyzed" (Myers, 2013). Grounded theory is an inductive research

method that is fundamentally different from the traditional experimental research methods described in Chapters 2 and 3. As demonstrated in Figure 11.1, when conducting experimental research, we normally start from a preformed theory, typically in the form of one or more hypotheses, we then conduct experiments to collect data and use the data to prove the theory. In contrast, grounded theory starts from a set of empirical observations or data and we aim to develop a well-grounded theory from the data. During the process of theory development, multiple rounds of data collection and analysis may be conducted to allow the underlying theory to fully emerge from the data (Myers, 1997; Corbin and Strauss, 2014). Therefore, some researchers refer to the theory generated using this method as the "reverse-engineered" hypothesis.

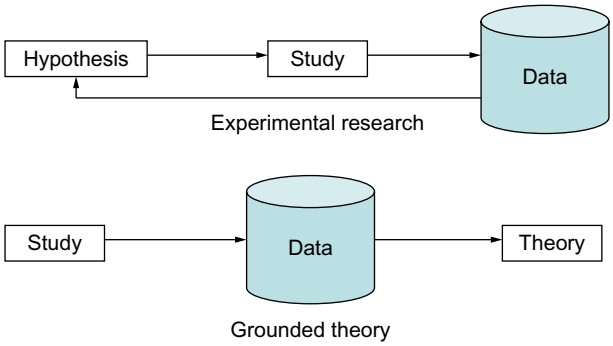

FIGURE 11.1

Experimental research compared with grounded theory.

Grounded theory can be applied to a variety of research methods discussed in this book such as ethnography (Chapter 9), case studies (Chapter 7), and interviews (Chapter 8). The major difference between qualitative research strategies that are mainly descriptive or exploratory and grounded theory is its emphasis on theory development from continuous interplay between data collection and data analysis.

Because grounded theory does not start from a preformed concept or hypothesis, but from a set of data, it is important for researchers to start the research process without any preconceived theoretical ideas so that the concepts and theory truly emerge from the data. The key to conducting successful grounded theory research is to be creative and have an open mind (Myers, 2013). Since grounded theory was first proposed in 1967, opinions on how to conduct research using grounded theory have diverged (Glaser, 1992; Strauss, 1987; Corbin and Strauss, 2014). The founders disagree on whether grounded theory can be formalized into a set of clear guidelines and procedures. Glaser believes that procedures are far too restrictive and may contradict the very basis of this method: creativity and an open mind. Even with the public disagreement, the procedures and guidelines proposed by Strauss and Corbin have been widely used in the field of social science, probably partly due to the fact that the procedure makes grounded theory more tangible and easier to implement.

We briefly introduce the procedures of grounded theory according to Corbin and Strauss (Corbin and Strauss, 2014).

The grounded theory method generally consists of four stages:

- open coding;
- development of concepts;
- grouping concepts into categories;
- formation of a theory.

In the open coding stage, we analyze the text and identify any interesting phenomena in the data. Normally each unique phenomenon is given a distinctive name or code. Given a piece of text to analyze, you would read through, trying to identify the patterns, opinions, behaviors, or other issues that sound interesting. Since you are not constrained by preestablished theories, frameworks, or concepts, you are open to all possibilities that reside in the data.

During this process, you need to find terms to describe the interesting instances that emerge from the data. Sometimes the participants may provide terms that describes the instances or key elements so vividly or accurately that you can borrow the term directly. Coding categories generated in this manner are called *in vivo* code. In vivo coding can help ensure that the concepts stay as close as possible to the participants' own words. These types of codes are largely adopted when using the grounded theory method. In one survey that the authors conducted on computer usage by children with Down syndrome, we borrowed many terms (e.g., curriculum integration) directly from parents' response and used them as low-level themes (Feng et al., 2010).

When the original text does not contain a key term to describe the instance of interest, the researcher will need to find an appropriate term to describe the instance. Those terms are called "researcher-denoted concepts." For example, if you read the following descriptions in the data, you may use the term "frustration" to describe the underlying theme of both responses:

> *My son just sits there and sobs when the computer does not do what he wants.*
> *He becomes irritated and keeps pushing the Enter button when the web page loads slowly.*

In the second stage, collections of codes that describe similar contents are grouped together to form higher level "concepts," which can then be grouped to form "categories" (the third stage). Definitions of the concepts and categories are often constructed during this phase of the analysis. The identification and definition of relationships between these concepts—a process often referred to as "axial coding" (Preece et al., 2015; Corbin and Strauss, 2014) is often a key step in this process. As analysis continues, we are constantly searching for and refining the conceptual construct that may explain the relationships between the concepts and categories (Glaser, 1978).

Although this description implies a linear process with well-defined phases, analyses might not be quite so clear-cut. The identification of new codes through open coding, the grouping of these codes into categories, and the definition of relationships

between these codes is a complex process involving the evolving construction of an understanding of the data. Iterative review of the data is often a key part of the process, as identification of new codes and categories might lead you to rereview documents from the perspective of codes identified in later documents. This rereview might also suggest multiple categorizations or types of relations between codes.

In the last stage, theory formulation, we aim at creating inferential and predictive statements about the phenomena recorded in the data. More specifically, we develop explicit causal connections or correlations between the concepts and categories identified in the previous stages. This process might be followed by *selective coding,* in which previously coded data might be revisited from the context of the emerging theory. Of course, further iteration and identification of open codes or axial codes is also possible.

A study of the issues involved in building information technology support for palliative care provides an instructive example of the use of emergent coding and grounded theory. Noting that palliative care differs significantly from other forms of medical care in a focus on the individual needs of each patient, Kuziemsky and colleagues conducted a grounded theory analysis of multiple data sources, including 50 hours of interviews with seven professionals (nurses, physicians, and counselors), patient charts, and research literature. Figure 11.2 provides an example of open and axial codes used in this analysis. This coding process was used to form a more detailed map of relationships between factors important to palliative care (Kuziemsky et al., 2007).

While conducting research using grounded theory, it is important to fully understand the advantages and limitations of this research method. Grounded theory obviously has a number of advantages. First, it provides a systematic approach to analyzing qualitative, mostly text-based data, which is impossible using the traditional experimental approach. Second, compared to the other qualitative research methods, grounded theory allows researchers to generate theory out of qualitative data that can be backed up by ample evidence as demonstrated in the thorough coding. This is one of the major attractions of the grounded theory and even novice users found the procedure intuitive to follow. Third, grounded theory encourages researchers to study the data early on and formulate and refine the theory through constant interplay between data collection and analysis (Myers, 2013).

On the other hand, the advantages of grounded theory can become disadvantages at times. It is not uncommon for novices to find themselves overwhelmed during the coding stage. The emphasis on detailed and thorough coding can cause researchers to be buried in details and feel lost in the data, making it difficult to identify the higher-level concepts and themes that are critical for theory formulation. In addition, theories developed using this method may be hard to evaluate. Unlike the traditional experimental approach in which the hypothesis is clearly supported or rejected by quantitative data collected through well-controlled, replicable experiments, grounded theory starts from textual information and undergoes multiple rounds of data collection and coding before the theory fully emerges from the data. The evaluation of the outcome depends on measures that are less direct, such as the chain of evidence between the

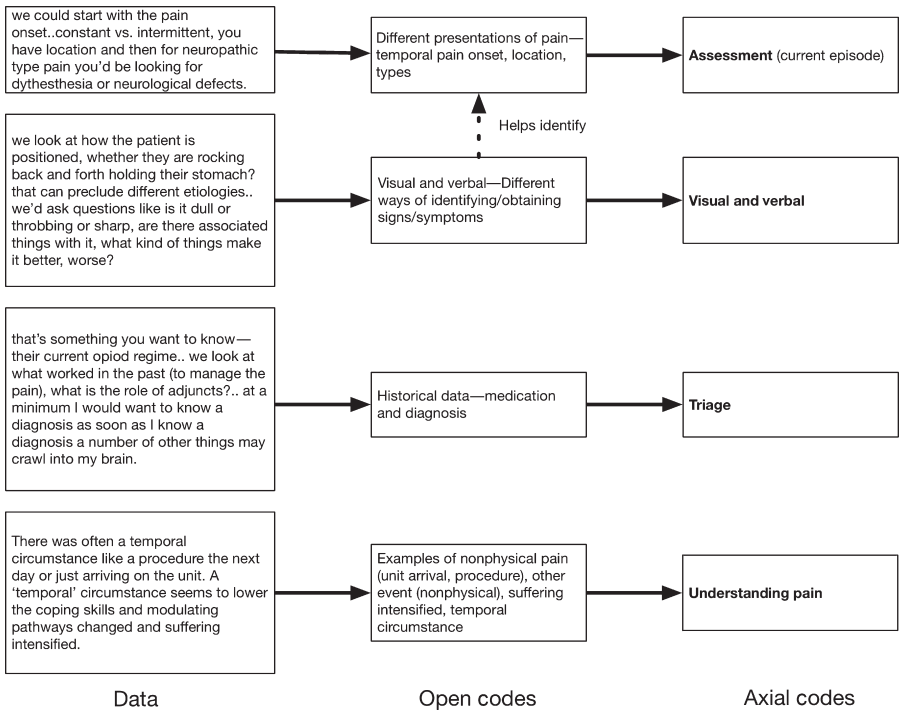

FIGURE 11.2

Example open and axial codes from a grounded theory analysis of issues relating to palliative care pain management. Note that the axial codes both abstract multiple open codes into more general categories and also (in the case of the arrow labelled "helps identify") describe relationships between the codes.

Adapted from Kuziemsky, C.E., et al., 2007. A grounded theory guided approach to palliative care systems design. International Journal of Medical Informatics 76, S141–S148.

finding and the data, the number of instances in the data that support the specific concept, and the familiarity of the researcher with the related topic. Lastly, the findings of the grounded theory approach may be influenced by the researchers' preconceived opinions and, therefore, may be subject to biases. In order to avoid these issues from happening, researchers should always keep in mind the key of this approach: being creative and open minded; listening to the data. When there is a gap between the concept and the data, additional data need to be collected to fill in the gap and tighten the linkage between the concept and the data. Due to these limitations, some researchers prefer to use grounded theory just as a coding technique, not as a theory generation method. For a detailed exploration of these and many other issues relating to the use of grounded theory in qualitative analysis, see "The SAGE Handbook of Grounded Theory" (Bryant and Charmaz, 2007) and Corbin and Strauss' classic text *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory* (Corbin and Strauss, 2014).

11.4.1.2 A priori coding and theoretical frameworks

Theoretical frameworks are commonly used in multiple stages of qualitative research (Corbin and Strauss, 2014). In the research design stage, theoretical frameworks can help you frame the research questions, decide on the specific research approach to adopt (i.e., survey, interview, focus group, etc.), and identify the concepts and questions to be included in each approach. When analyzing text information, theoretical frameworks can help you identify the major categories and items that need to be coded and explain the findings of your research. Therefore, at the beginning of a research project, it is important to study the research literature and find out whether there is any theoretical framework related to the research topic that you are investigating.

For example, suppose you interview a number of senior citizens to examine the major difficulties that they experience when using computers. One question you would like to answer is the underlying cause of those difficulties. You know a large proportion of the difficulties can be attributed to the gradual decline of human capabilities. According to well-established literature, human capabilities can be grouped into three major categories: cognitive, physical, and perceptual abilities. You can use those three types of capability as the high-level categories of your coding scheme and try to group the participants' responses in each of those three categories.

In the HCI field, theoretical frameworks are also called taxonomies. Numerous taxonomies have been developed to help guide research and understand the data collected through various user studies. One example of the earlier taxonomies proposed is about the types of task that users conduct (Norman, 1991). By grouping tasks into categories, such as "structured and unstructured" or "regular and intermittent," and summarizing the different nature and requirements of each type of task, researchers and designers can study the interaction in a consistent way and make easier connections between different aspects of the result. For a comprehensive discussion on task analysis, see (Courage et al., 2007). Another widely cited taxonomy in the HCI field groups human errors into mistakes and slips (Norman, 2013). Slips can be further categorized into capture errors, description errors, data-driven errors, associative-activation errors, loss-of-activation errors, and mode errors. Each type of slip has different causes, and different design techniques can be used to help prevent, detect, and recover from those errors. Well-studied and validated taxonomies can provide great insights for identifying the potential categories to be included for coding.

In both emergent coding and a priori coding, different coding techniques may be adopted depending on the nature of the data and the study context. Examples include magnitude coding, in which codes are associated with qualitative or quantitative assessments of the strength or frequency of the concept; process coding, which uses gerunds ("-ing" verbs) to identify actions; and a variety of affective coding methods focused on emotions and values. Saldaña (Saldaña, 2012) provides an in-depth catalog of different coding approaches and methodologies.

11.4.1.3 Building a code structure

After the key coding items are identified, they can be organized and presented in a code list (also called a "nomenclature" or a "codebook"). A nomenclature is a list of

numbered categories intended to represent the full array of possible responses to a specific question (Lyberg and Kasprzyk, 1997). For studies using theoretical frameworks, the codes will come from the categories and concepts identified by the theory. Emergent coding, however, means that the codes are not identified in advance—the list will emerge as new concepts of interest are found in the source material. A code list is normally built into a hierarchical structure, containing multiple levels, each level representing concepts with increasing amounts of detail. Building a code structure is not an easy task. It requires both extensive knowledge of the existing theories and literature and a deep understanding of the data collected. Many times, the analyst needs to make compromises between the theoretical framework and the practical aspects of the study.

Figure 11.3 demonstrates an example of a code structure generated by Feng et al. (2010) when investigating computer usage behaviors by children with Down syndrome. The researchers adopted a mixture of emergent coding and a priori coding

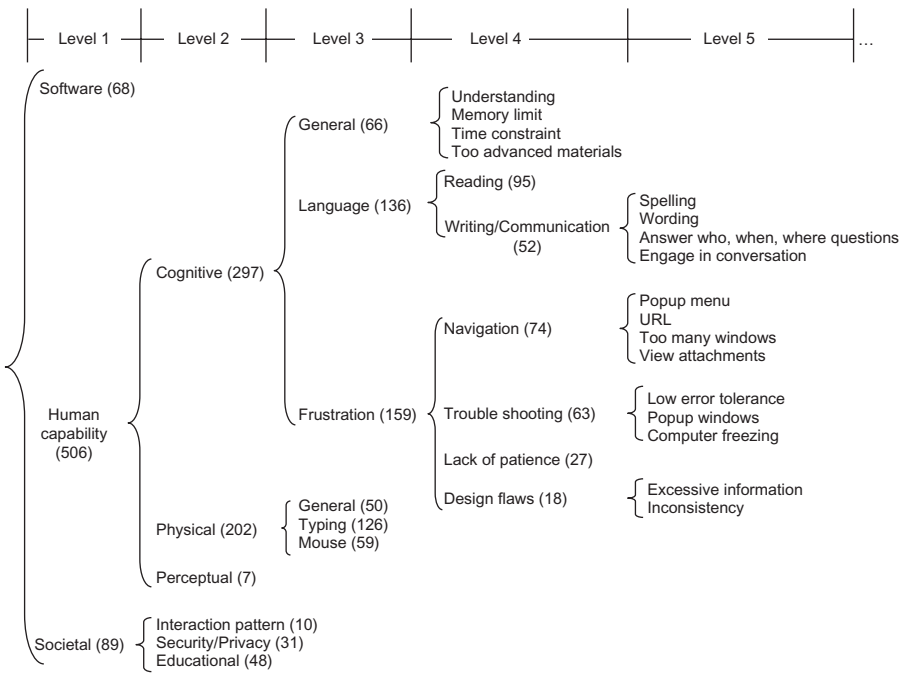

FIGURE 11.3

Example code structure using both emergent coding and a priori coding about difficulties experienced by children with Down syndrome when using computers or computer-related devices (Numbers in parentheses represent the number of children whose parents reported the particular type of difficulty. Some parents reported in more than one subcategories so the numbers do not necessarily add up to the total number in the parent category.)

Excerpted from Feng, J., et al., 2010. Computer usage by children with down syndrome: challenges and future research. ACM Transactions on Accessible Computing 2 (3), 32. Copyright ACM.

when trying to identify the key concepts and categories. Most of the code items at level three, four, and five were identified through emergent coding, and so were the three categories listed under "Societal difficulties." In contrast, the three categories under "Human capabilities" were derived from existing theory from psychology and behavioral science.

11.4.2 CODING THE TEXT

When the data set is not large, which is typically true for interviews, focus groups, or observations, it is recommended to read the text from beginning to end before starting to do any coding. During the first round of reading, you may find interesting issues and feel the urge to write among the text or in the margins. Those activities should wait until you start the coding. The purpose of this first round of reading is to immerse you into the life and experience of the participants and get a general, unbiased idea of the data set before focusing on any specific aspects. After this first read-through, you should be ready to dive in and start coding.

Inexperienced coders may find it difficult to identify anything interesting (or anything that is worth being coded) in the data, especially when the coding category is not established and they are doing open coding to identify coding categories or themes. Other coders may experience the other end of the scale: they may feel that the data is so rich that they need to code almost every word or phrase. Eventually they may be overwhelmed by the large number of coding items that they are trying to document. They may be distracted by the less important or even trivial coding items and fail to identify the most interesting or informative patterns in the data. In order to avoid both situations, we recommend the following steps for coding:

1. Look for specific items.
2. Ask questions constantly about the data.
3. Making comparisons constantly at various levels.

We'll discuss these steps in the following sections.

11.4.2.1 Look for key items

While coding the data, specific types of "statements" are more likely to carry valuable information. A partial list of such statements is given in Table 11.2. In fact, these categories might prove useful as codes on their own!

Objectives deliver important information. A user's computer usage behavior and interaction style is largely affected by the objectives that they want to achieve. If a user uses a specific application just for entertainment, it may be unrealistic to expect the user to devote a substantial amount of time to learning how to use the application. It would be totally different if the application is a critical tool at work.

Words, phrases, and sentences that describe actions are also important. They tell you what the users do with the specific application or technique. They also tell you what functions are frequently used and what are less frequently used. Once you detect an action code in the data, you can follow up on that and examine whether the user

Table 11.2 Some Examples of Statements to Look for While Coding

Statement	Examples
Objectives	Use computers for educational purposes
Actions	Enter a password, chat online
Outcomes	Success or failure, whether the objective is achieved
Consequences	Files unintentionally deleted, a specific application abandoned
Causes	Limited memory, dated equipment
Contexts	User is computer savvy, user works with classified information
Strategies	Avoid specific tasks, multimodal interaction

described the outcome of the action. Was the action successfully completed? Did the action completely fail? Was the action partially completed? Whenever an action is not completely successful, you may want to pursue the consequences or costs of the unsuccessful action: Is the consequence highly detrimental? Does it cause the user to lose several days of work? Does it prevent the user from completing some tasks on time? Is it a minor nuisance or is it so frustrating that the user decides to abandon the action?

Causes are also associated with failed actions. Whenever an action completely or partially fails, it is worth pursuing the causes of the failure. Does the failure trace back to the user or the application? If it is caused by the user, what kinds of capability are involved? Is it due to cognitive overload? Is it due to lack of attention? Is it due to physical or perceptual limitations? Or is it due to the interaction between two or more of those factors. Statements about the context of the interaction or usage are also important. Different types of user may report different satisfaction levels for the same application with similar performance measures because the comparison context is drastically different (Sears et al., 2001). Finally, descriptions of interaction styles and strategies are also valuable information that is hard to examine during empirical lab-based studies.

11.4.2.2 Ask questions about the data

A good way to help detect interesting patterns and connections in data is to constantly ask questions about the data. In Section 11.4.2.1, we listed a series of questions that you can ask once you identify an interesting action in the data. Those questions can be related to the specific action, its outcome, and its consequence, as well as the causes of failed actions. Most of those questions are practical questions that may help you identify interaction challenges and design flaws.

Corbin and Strauss (Corbin and Strauss, 2014) discussed the art of asking questions in a larger context with the primary objective of theory development. They proposed four types of questions and two of them are particularly important during the analysis phase: sensitizing questions and theoretical questions. Sensitizing questions help coders better understand the meaning of the data: What is happening here? What did the user click? How did the user reach the specific web page? Theoretical questions help the researchers make connections between concepts and categories: What is the relationship between two factors? How does the interaction change over time?

11.4.2.3 Making comparisons of data

Both during the coding process and the stage afterwards to interpret the results, you are encouraged to make comparisons at multiple levels. First, you can compare instances under different coding categories. For example, if you are investigating the difficulties that older people experience when using computers, you can compare the frequency with which each capability (physical, cognitive, perceptual) is reported. You can also compare the degree of impact between different capabilities.

Second, you can compare the results between different participant groups. You may find that the capabilities and computer usage behaviors vary substantially among the users. You can further investigate this diversity via different dimensions: Is the diversity related to age, educational background, or community and family support? To answer these questions, you need to subdivide the data set and compare the results among subsets.

Third, you can compare the findings in your data to previously reported literature. Do your findings align with the existing literature or is it contradictory? If your findings differ from existing literature, can you explain why? Is the existing literature incorrect? Or is the difference caused by a different context? Sometimes the need to compare your findings with a related population or tasks may facilitate you to conduct additional studies to collect more data. For example, if you observe some interesting computer usage behavior in children with autism, you may want to conduct the same study for neurotypical children to investigate whether there is a difference between the neurotypical children and children with Autism.

11.4.2.4 Recording the codes

When you find an item in the content that you wish to describe with a code, you should note exactly what you are coding and which codes you are assigning. The "what" should provide enough detail to unambiguously identify the relevant content—you might quote specific paragraphs, sentences, or phrases from a textual document or identify start and end points for time intervals in an audio recording. The code(s) that you are assigning will come from your list of codes. Note that coders often find that they want to describe a given item with multiple codes.

A variety of strategies can be used to record codes. One low-tech approach might involve marking up text with comment and highlighting tools from a word processor. Another possibility might be to use a spreadsheet with columns for the identification of the item being coded and for the codes being assigned.

A number of commercial and academic software packages provide dedicated support for text analysis, including tools for creating codebooks, coding documents, and searching and querying for material associated with given codes. Other tools support automated text content analysis via features for searching, counting, sorting, and conducting basic statistics. Examples of commonly used text analysis software include SAS (https://www.sas.com), GATE (the General Architecture for Text Engineering) (https://gate.ac.uk), and Carrot2 (http://project.carrot2.org).

11.4.2.5 Iterating and refining

Qualitative coding leads to the construction of an evolving conceptual framework. As you examine raw data and assign codes to elements of that data, you are in effect organizing the components and constructing an understanding that will grow and change as you continue. For emergent coding efforts, the addition of new codes is the emergence of your understanding. However, even theoretically informed efforts may find that a deeper appreciation of the data leads to the realization that the initial framework is not quite adequate or correct. If this happens, you may wish to add codes to your codebook, and to reconsider previously coded material in the light of these new codes. This iterative extension of the codebook and rereview of material can be time consuming, but it does reflect the evolving nature of your understanding.

11.4.3 ENSURING HIGH-QUALITY ANALYSIS

Qualitative data analysis is not objective. During the data-coding process, a human researcher makes a series of decisions regarding the interpretation of individual observations: Which category does this item belong in? Are these items really members of the same group or should they be separated? No matter how expert the judgment of the individual making these decisions, the possibility of some conscious or unconscious bias exists. Given the inherent fallibility of human researchers, how can we increase our confidence in the results of qualitative analysis? More specifically, how can we make our qualitative analysis *valid* and *reliable*?

Before we can answer that question, we must be clear on what we mean by these terms. In terms of qualitative research, *validity* means that we use well-established and well-documented procedures to increase the accuracy of findings (Creswell, 2013). More strictly speaking, validity examines the degree to which an instrument measures what it is intended to measure (Wrench et al., 2013). *Reliability* refers to the consistency of results (Creswell, 2013): if different researchers working on a common data set come to similar conclusions, those conclusions are said to be reliable.

Ensuring reliability and validity of qualitative HCI research is a challenge. For additional guidance on improving the rigor of your qualitative research—and, indeed, on all aspects of qualitative HCI—see the monograph *Qualitative HCI Research: Going Behind the Scenes* (Blandford et al., 2016).

11.4.3.1 Validity

Validity is a very important concept in qualitative HCI research in that it measures the accuracy of the findings we derive from a study. There are three primary approaches to validity: face validity, criterion validity, and construct validity (Cronbach and Meehl, 1955; Wrench et al., 2013).

Face validity is also called content validity. It is a subjective validity criterion that usually requires a human researcher to examine the content of the data to assess whether on its "face" it appears to be related to what the researcher intends to

measure. Due to its high subjectivity, face validity is more susceptible to bias and is a weaker criterion compared to construct validity and criterion validity. Although face validity should be viewed with a critical eye, it can serve as a helpful technique to detect suspicious data in the findings that need further investigation (Blandford et al., 2016).

Criterion validity tries to assess how accurate a new measure can predict a previously validated concept or criterion. For example, if we developed a new tool for measuring workload, we might want participants to complete a set of tasks, using the new tool to measure the participants' workload. We also ask the participants to complete the well-established NASA Task Load Index (NASA-TLX) to assess their perceived workload. We can then calculate the correlation between the two measures to find out how the new tool can effectively predict the NASA-TLX results. A higher correlation coefficient would suggest higher criterion validity. There are three subtypes of criterion validity, namely predictive validity, concurrent validity, and retrospective validity. For more details regarding each subtype—see Chapter 9 "Reliability and Validity" in Wrench et al. (2013).

Construct or factorial validity is usually adopted when a researcher believes that no valid criterion is available for the research topic under investigation. Construct validity is a validity test of a theoretical construct and examines "What constructs account for variance in test performance?" (Cronbach and Meehl, 1955). In Section 11.4.1.1 we discussed the development of potential theoretical constructs using the grounded theory approach. The last stage of the grounded theory method is the formation of a theory. The theory construct derived from a study needs to be validated through construct validity. From the technical perspective, construct or factorial validity is based on the statistical technique of "factor analysis" that allows researchers to identify the groups of items or factors in a measurement instrument. In a recent study, Suh and her colleagues developed a model for user burden that consists of six constructs and, on top of the model, a User Burden Scale. They used both criterion validity and construct validity to measure the efficacy of the model and the scale (Suh et al., 2016).

In HCI research, establishing validity implies constructing a multifaceted argument in favor of your interpretation of the data. If you can show that your interpretation is firmly grounded in the data, you go a long way towards establishing validity. The first step in this process is often the construction of a database (Yin, 2014) that includes all the materials that you collect and create during the course of the study, including notes, documents, photos, and tables. Procedures and products of your analysis, including summaries, explanations, and tabular presentations of data can be included in the database as well.

If your raw data is well organized in your database, you can trace the analytic results back to the raw data, verifying that relevant details behind the cases and the circumstances of data collection are similar enough to warrant comparisons between observations. This linkage forms a chain of evidence, indicating how the data supports your conclusions (Yin, 2014). Analytic results and descriptions of this chain of evidence can be included in your database, providing a roadmap for further analysis.

A database can also provide increased reliability. If you decide to repeat your experiment, clear documentation of the procedures is crucial and careful repetition of both the original protocol and the analytic steps can be a convincing approach for documenting the consistency of the approaches.

Well-documented data and procedures are necessary, but not sufficient for establishing validity. A very real validity concern involves the question of the confidence that you might have in any given interpretive result. If you can only find one piece of evidence for a given conclusion, you might be somewhat wary. However, if you begin to see multiple, independent pieces of data that all point in a common direction, your confidence in the resulting conclusion might increase. The use of multiple data sources to support an interpretation is known as data source *triangulation* (Stake, 1995). The data sources may be different instances of the same type of data (for example, multiple participants in interview research) or completely different sources of data (for example, observation and time diaries).

Interpretations that account for all—or as much as possible—of the observed data are easier to defend as being valid. It may be very tempting to stress observations that support your pet theory, while downplaying those that may be more consistent with alternative explanations. Although some amount of subjectivity in your analysis is unavoidable, you should try to minimize your bias as much as possible by giving every data point the attention and scrutiny it deserves, and keeping an open mind for alternative explanations that may explain your observations as well as (or better than) your pet theories.

You might even develop some alternative explanations as you go along. These alternatives provide a useful reality check: if you are constantly re-evaluating both your theory and some possible alternatives to see which best match the data, you know when your theory starts to look less compelling (Yin, 2014). This may not be a bad thing—rival explanations that you might never find if you cherry-picked your data to fit your theory may actually be more interesting than your original theory. Whichever explanations best match your data, you can always present them alongside the less successful alternatives. A discussion that shows not only how a given model fits the data but how it is a better fit than plausible alternatives can be particularly compelling.

Well-documented analyses, triangulation, and consideration of alternative explanations are recommended practices for increasing analytic validity, but they have their limits. As qualitative studies are interpretations of complex datasets, they do not claim to have any single, "right" answer. Different observers (or participants) may have different interpretations of the same set of raw data, each of which may be equally valid. Returning to the study of palliative care depicted in Figure 11.2, we might imagine alternative interpretations of the raw data that might have been equally valid: comments about temporal onset of pain and events might have been described by a code "event sequences," triage and assessment might have been combined into a single code, etc. Researchers working on qualitative data should take appropriate measures to ensure validity, all the while understanding that their interpretation is not definitive.

11.4.3.2 Reliability

The ambiguous data that is the focus of content analysis exemplifies many of the reliability challenges presented by qualitative data analysis. The same word may have different meanings in different contexts. Different terms or expressions may suggest the same meaning. The data may be even more ambiguous when it comes to the interpretation of body language, facial expression, gestures, or art work. The same people may interpret the same gesture differently after viewing it at different times. In many studies, the data set is very large and multiple coders may code different subsets of the data. Due to the nature of content analysis, it is more vulnerable to biases and inconsistencies than the traditional quantitative approach. Therefore, it is particularly important to follow specific procedures during the coding process and use various measures to evaluate the quality of the coding. The ultimate goal of reliability control is to ensure that different people code the same text in the same way (Weber, 1990).

Reliability checks span two dimensions: stability and reproducibility. Stability is also called *intracoder reliability*. It examines whether the same coder rates the data in the same way throughout the coding process. In other words, if the coder is asked to code the same data multiple times, is the coding consistent time after time? If the coder produces codes that shows 50% in category A, 30% in category B, and 20% in category C the first time; then 20% in category A, 20% in category B, and 60% in category C the second time, the coding is inconsistent and the intracoder reliability is very low.

In the context of content analysis, intercoder reliability is widely adopted to measure reproducibility. It examines whether different coders code the same data in a consistent way. In other words, if two or more coders are asked to code the same data, is their coding consistent? In this case, if one coder produces codes that shows 50% in category A, 30% in category B, and 20% in category C; while the other coder produces codes that show 20% in category A, 20% in category B, and 60% in category C, then the coding is inconsistent and the intercoder reliability is very low.

A further step in demonstrating reliability might use multiple coders specifically chosen for differences in background or theoretical perspectives, leading to a theoretical triangulation (Stake, 1995). If individuals with substantially different intellectual frameworks arrive at similar conclusions, those results may be seen as being very reliable.

In order to achieve reliable coding both from the same coder and among multiple coders, it is critical to develop a set of explicit coding instructions at the beginning of the coding process. All of the coders need to be trained so that they fully understand the instructions and every single coding item. The coders then test code some data. The coded data is examined and reliability measures are calculated. If the desired reliability level is achieved, the coders can start the formal coding. If the desired reliability level is not achieved, measures must be taken to improve reliability. These measures might include retraining and recoding the data used in the test coding. Alternatively, the coders might use a discussion of disagreements to determine how coding should be conducted, and revise the codebook and coding instructions to reflect the new consensus. After the formal coding process starts, it is important to

conduct reliability checks frequently so that inconsistent coding can be detected as early as possible.

One of the commonly used reliability measures is the percentage of agreement among coders, calculated according to the following equation:

$$\%\text{agreement} = \frac{\text{the number of cases coded the same way by multiple coders}}{\text{the total number of cases}}$$

When analyzing a survey on software and technology for children with autism, Putnam and Chong (2008) coded the data independently and reported a 94% agreement between the two coders, which is quite a satisfactory level. However, the percentage agreement approach does have a limitation: it does not account for the fact that several coders would agree with each other for a certain percentage of cases even when they just code the data by chance. Depending on the specific feature of the coding, that percentage may be quite substantial.

To address this limitation, you can adopt other measures such as Cohen's Kappa (Cohen, 1960), which rates interrater reliability on a scale from 0 to 1, with 0 meaning that the cases that are coded the same are completely by chance and 1 meaning perfect reliability. Kappa is calculated by the following equation:

$$K = \frac{P_a - P_c}{1 - P_c}$$

where P_a represents the percentage of cases on which the coders agree and P_c represents the percentage of agreed cases when the data is coded by chance.

Suppose we conduct a survey of senior citizens and ask them to describe the primary causes of the difficulties that they encounter when using computers. We identify three major categories of causes: difficulties due to physical capabilities, difficulties due to cognitive capabilities, and difficulties due to perceptual capabilities. Two coders code the data independently. Their coding results are summarized in an agreement matrix as illustrated in Table 11.3. The diagonal line from top left shows the percentages of cases on which the coders agreed. For example, the number of cases that both coders coded under the "physical difficulty" category accounts for 26% of the total number of cases. The other cells contain the cases on which the two coders disagreed (i.e., 7% of the cases were coded under "physical difficulties" by the first coder and under "cognitive difficulties" by the second coder). The "marginal

Table 11.3 The Distribution of Coded Items Under Each Category by Two Coders (Agreement Matrix)

		Coder 2			
		Physical	**Cognitive**	**Perceptual**	**Marginal total**
Coder 1	**Physical**	0.26 (0.14)	0.07 (0.08)	0.04 (0.15)	0.37
	Cognitive	0.04 (0.07)	0.12 (0.04)	0.01 (0.07)	0.17
	Perceptual	0.09 (0.18)	0.02 (0.10)	0.35 (0.18)	0.46
	Marginal total	0.39	0.21	0.40	1.00

totals" are calculated by adding up the values in each row or column. The "marginal total" values always add up to one. The value in parentheses in each cell represents the expected percentage agreement when the data is coded by chance, calculated by multiplying the marginal totals of the corresponding row and column (i.e., the expected percentage agreement for (physical, physical) is $0.37 \times 0.39 = 0.14$).

Based on the data provided by Table 11.3, we can compute the value of P_a as:

$$P_a = 0.26 + 0.12 + 0.35 = 0.73$$

The value of P_c is computed by adding the expected percentage agreement (in parentheses on the diagonal):

$$P_c = 0.14 + 0.04 + 0.18 = 0.36$$

Therefore,

$$K = \frac{0.73 - 0.36}{1 - 0.36} = 0.58$$

A well-accepted interpretation of Cohen's Kappa is that a value above 0.60 indicates satisfactory reliability. Table 11.4 summarizes a more detailed interpretation of Cohen's Kappa (Landis and Koch, 1977; Altman, 1991). When the value of Kappa is below 0.60, the reliability of the analysis is questionable.

Table 11.4 Interpretation of Cohen's Kappa

Interpretation	Kappa range
Poor or slight agreement	$K \leq 0.20$
Fair agreement	$0.20 < K \leq 0.40$
Moderate agreement	$0.40 < K \leq 0.60$
Satisfactory agreement	$0.60 < K \leq 0.80$
Near-perfect agreement	$K > 0.80$

In addition to the percentage agreement and Cohen's Kappa, there are several other coefficients that measure coder agreement, such as Osgood's coefficient (also named CR) proposed by Osgood (1959) and the S coefficient proposed by Bennett et al. (1954). Hallgren (2012) provided a more detailed tutorial on Cohen's Kappa and related measures. For detailed discussions of the differences among the agreement measures, see Krippendorff (2004) or Artstein and Poesio (2008).

The process of achieving high interrater reliability often involves multiple iterations, as low initial reliability might lead to changes in codebooks and/or instructions. Once acceptable reliability has been achieved on a subset of the data, coders are presumed to be reliable and can proceed independently without further checks. Whenever possible, having multiple coders review all documents at a high-level of reliability is preferred, but in some cases resource limitations may require multiple coding of only a subset of the data.

11.4.3.3 Subjective versus objective coders

You should be aware of the advantages and disadvantages of using subjective or objective coders and their impact on coding reliability. When the coders are the same people who developed the coding scheme, and in many cases they also design the study and collect the data, they are called *subjective* or *inside* coders. When the coders are not involved in the design of the study, the data collection, or the development of the coding scheme, they are called *objective* or *outside* coders.

There are pros and cons of both approaches. Because subjective coders are usually the researchers themselves, they know the literature well and have substantial knowledge and expertise in the related topic. That knowledge and specialty can help them understand the terms and concepts provided by participants and detect the underlying themes in the text. They also require minimal training since they developed the coding scheme themselves. However, the fact that they have already worked so closely with the data becomes a disadvantage during the actual coding. The preacquired knowledge may constrain their abilities to think beyond the established concepts in their mind. Sometimes they may form hidden meanings of the coding without being aware of it. The consequence is that the reliability reported by subjective coders may be inflated (Krippendorff, 1980).

On the contrary, objective coders usually do not have preacquired knowledge of the subject and, therefore, may be more open to potential instances in the data. The reliability reported by objective coders is less likely to be inflated. However, their lack of domain knowledge and expertise may also hinder their ability to accurately understand the data and detect interesting instances. In addition, objective coders usually need a substantial amount of training and the entire process can be very costly.

In practice, it is very common for studies to use subjective coders for content analysis and this approach is usually considered acceptable as long as the appropriate procedure is followed and reported, along with the reliability measures.

11.5 ANALYZING MULTIMEDIA CONTENT

Multimedia data has become prevalent in our daily life thanks to the rapid advances in affordable portable electronic devices and storage technologies. Researchers can collect a large quantity of image, audio, and video data at fairly low cost. Multimedia information such as screen shots, cursor movement tracks, facial expressions, gestures, pictures, sound, and videos provide researchers an amazingly rich pool of data to study how users interact with computers or computer-related devices.

Multimedia information also presents substantial challenges for data analysis. In order to find interesting patterns in the interactions, the image, audio, and video data need to be coded for specific instances (i.e., a specific gesture, event, or sound). Without the support of automated tools, the researcher would have to manually go through hours of audio or video recordings to identify and code the instances of

specific interest. This process can be extremely time-consuming, tedious, and in many cases, impractical.

The basic guidelines for analyzing text content also apply to multimedia content. Before you start analyzing the data, you need to study the literature and think about the scope, context, and objective of your study. You need to identify the key instances that you want to describe or annotate. After the analysis, you need to evaluate the reliability of the annotation. If a manual annotation approach is adopted, it may be a good idea to select a subset of the entire data set for analysis due to high labor cost. For example, Peltonen et al. (2008) picked eight days of data from a study that lasted for 1 month. They first automatically partitioned the video footage into small "sessions," then manually coded the information in which they were interested (the duration of interaction, the number of active users, and the number of passive bystanders).

Another application domain related to multimedia content analysis is the online search of media content. There is a huge amount of images, videos, and audios on the web. Users frequently go online to search for images, videos, or audio materials. Currently, most multimedia search is completed by text-based retrieval, which means that the multimedia materials have to be annotated or labeled with appropriate text. So far, annotation can be accomplished through three approaches: manual annotation, partially automated annotation, and completely automated annotation.

Considering the huge amount of information that needs to be annotated, the manual approach is extremely labor intensive. In addition, it can also be affected by the coder's subjective interpretation. The completely automated approach is less labor intensive. However, due to the substantial semantic gap between the low-level features that we can currently automatically extract and the high-level concepts that are of real interest to the user, existing automatic annotation applications are highly error prone (i.e., many images that have nothing to do with cats may be annotated with "cat" using this automatic annotation). A more recent development in this field is the partially automated approach. Human coders manually annotate a subset of the multimedia data. Then the manually coded data is used to train the application to establish the connection between the low-level features and the high-level concept. Once a concept detector is established, the detector can be used to automatically annotate the rest of the data (Rui and Qi, 2007). The same approach can be applied to images and video and audio clips.

The techniques for multimedia content analysis are built on top of multiple domains including image processing, computer vision, pattern recognition and graphics. One of the commonly adopted approaches used by all those fields is machine learning. The specific algorithms or techniques of multimedia content analysis are still seeing dramatic advances. For more detailed information on those topics, see publications in the related fields (Hanjalic et al., 2006; Sebe et al., 2007; Divakaran, 2009; Ohm, 2016). The specific applications that are particularly interesting to the HCI field include action recognition and motion tracking (Zhu et al., 2006; Vondrak et al., 2012), body tracking (Li et al., 2006), face recognition, facial expression analysis (Wu et al., 2006; Wolf et al., 2016), gesture recognition (Argyros and Lourakis, 2006), object classification and tracking (Dedeoğlu et al., 2006; Guo et al., 2015),

and voice activity detection (Xue et al., 2006). A substantial number of studies have focused on automatic annotation and management of images.

In addition to the automatic annotation applications, a number of other tools have been developed to facilitate the process of multimedia content analysis. Dragicevic et al. (2008) developed a direct manipulation video player that allows a video analyst to directly drag and move the object of interest in the video to specific locations along their visual trajectory. Wilhelm et al. (2004) developed a mobile media metadata framework that enables image annotation on a mobile phone as soon as a picture is taken. The unique feature of this system is that it guesses the content of the picture for the purpose of reducing the amount of text entry needed during the annotation. Kandel et al. (2008) proposed the PhotoSpread system, which allows users to organize and analyze photos and images via an easy-to-use spreadsheet with direct manipulation functions. Applications that support content visualization for easy data sharing and analysis have also been developed (Cristani et al., 2008). The ChronoViz tool supports playback and review of multiple, synchronized streams of multimedia data (Fouse et al., 2011).

Techniques for automatic annotation still need substantial advancements in order to achieve reliable coding. The applications to facilitate manual coding have shown promising results but improvements are also needed to improve the usability and reliability of those systems.

11.6 SUMMARY

Text, multimedia, and other qualitative data are important sources of information for HCI researchers and practitioners. The procedure and techniques commonly used to analyze qualitative data are quite different from those applied to the analysis of quantitative data. Probably the most unique characteristic of content analysis is that it involves human coding. The absence of numeric data and direct measures makes qualitative data analysis more susceptible to biased interpretation or subjective manipulation. Therefore, it is critical to adopt well established procedures and techniques to ensure high-quality analysis that is both valid and reliable. Although there is disagreement regarding its implementation process and guidelines, grounded theory is widely used for qualitative data analysis. The major difference between grounded theory and other qualitative research strategies is its emphasis on theory development in continuous interplay between data collection and data analysis.

When analyzing text content, we need to develop a set of coding categories that accurately summarizes the data or describes the underlying relationships or patterns buried in the data. Depending on the specific context of the research question, a priori coding or emergent coding may be used to generate the coding categories. In order to produce high-quality coding, multiple coders are usually recommended to code the data. During the coding process, the coders should constantly look for statements likely to carry valuable information, ask questions about the data, and make

comparisons at various levels. Reliability control measures such as Cohen's Kappa should be calculated and evaluated throughout the coding process. Cohen's Kappa at or above 0.60 indicates satisfactory intercoder reliability.

The basic guidelines for analyzing text content also apply to multimedia content. Due to the special nature of multimedia data, the analysis can be much more labor-intensive than for text data if a completely manual annotation procedure is adopted. In order to address that challenge, a number of techniques have been developed to assist the annotation of multimedia data. To date, the completely automated annotation techniques are highly error prone. Applications to facilitate manual coding have shown promising results and may serve as a useful tool for analyzing multimedia data.

DISCUSSION QUESTIONS

1. What is the goal of qualitative analysis?
2. What are the stages of qualitative analysis?
3. What is content analysis?
4. What are the major types of content?
5. What do you need to consider before starting content analysis?
6. What is the difference between a priori coding and emergent coding?
7. What is grounded theory?
8. How does grounded theory differ from the traditional empirical research approach?
9. What are the four stages of grounded theory?
10. What is in vivo code?
11. What are the advantages and limitations of grounded theory?
12. What are the benefits of using theoretical frameworks when coding qualitative data?
13. What is a nomenclature/code book?
14. What is the procedure for analyzing text information?
15. What are the key items to look for while coding?
16. What is the meaning of ‘validity’ in qualitative analysis?
17. What is the meaning of ‘reliability’ in qualitative analysis?
18. What are the three primary types of validity in qualitative analysis?

19. What can you do to improve the validity of the findings of a HCI study?
20. Why do you need to conduct reliability checking during and after the coding process?
21. What is "stability" in the context of a reliability check?
22. What is "reproducibility" in the context of a reliability check?
23. What is the formula for computing Cohen's Kappa?
24. How do you interpret a specific value of Cohen's Kappa?
25. What is the difference between intracoder reliability and intercoder reliability?
26. What is the advantage and disadvantage of using a subjective coder?
27. What is the advantage and disadvantage of using an objective coder?
28. Why is analyzing multimedia content difficult?
29. How does the partially automated annotation method work?

RESEARCH DESIGN EXERCISE

You interview 50 children between the ages of 8 and 15 to study their computer usage behavior. During the data analysis, you find that the objective of using computers can be grouped into three categories: educational, communication, and entertainment. Two coders independently code the data and the agreement of their coding regarding computer usage objective is summarized in Table 11.5. Answer the following questions based on the agreement table:

1. Develop an agreement matrix. (Hint: You need to compute marginal totals for each row and column and the expected percentage agreement for each cell.)
2. Calculate Cohen's Kappa.
3. Discuss the result and determine whether the coding is reliable.

Table 11.5 Children's Computer Usage Objectives Coding Agreement

		Coder 2		
		Education	**Communication**	**Entertainment**
Coder 1	Education	0.49	0.05	0.02
	Communication	0.03	0.11	0.01
	Entertainment	0.04	0.02	0.23

REFERENCES

Altman, D.G., 1991. Practical Statistics for Medical Research. Chapman and Hall, London, England.

Argyros, A.A., Lourakis, M.I.A., 2006. Vision-based interpretation of hand gestures for remote control of a computer mouse. In: Huang, T.S., Sebe, N., Lew, M.S., et al. (Eds.), Computer Vision in Human-Computer Interaction: ECCV 2006 Workshop on HCI, Graz, Austria, May 13. Proceedings. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 40–51.

Artstein, R., Poesio, M., 2008. Inter-coder agreement for computational linguistics. Computational Linguistics 34 (4), 555–596.

Bennett, E.M., et al., 1954. Communications through limited-response questioning. Public Opinion Quarterly 18 (3), 303–308.

Berelson, B., 1952. Content Analysis in Communication Research. Free Press, Glencoe, IL.

Blandford, A., et al., 2016. Qualitative HCI research: going behind the scenes. Synthesis Lectures on Human-Centered Informatics 9 (1), 1–115.

Blumer, H., 1969. Symbolic Interactionism: Perspective and Method. Prentice-Hall, Englewood Cliffs, NJ.

Bryant, A., Charmaz, K., 2007. The SAGE Handbook of Grounded Theory. Sage Publications, Los Angeles, CA.

Cohen, J., 1960. A coefficient of agreement for nominal scales. Educational and Psychological Measurement 20 (1), 37–46.

Corbin, J., Strauss, A.L., 2014. Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory, fourth ed. Sage Publications, Los Angeles, CA.

Courage, C., et al., 2007. Task analysis. In: Sears, A., Jacko, J. (Eds.), The Human Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications. Lawrence Erlbaum Associates, New York.

Creswell, J.W., 2013. Research Design: Qualitative, Quantitative, and Mixed Methods Approaches. Sage Publications, Thousand Oaks, CA.

Cristani, M., et al., 2008. Content visualization and management of geo-located image databases. In: CHI'08 Extended Abstracts on Human Factors in Computing Systems. ACM, Florence, Italy, pp. 2823–2828.

Cronbach, L., Meehl, P., 1955. Construct validity in psychological tests. Psychological Bulletin 52, 22.

Dedeoğlu, Y., et al., 2006. Silhouette-based method for object classification and human action recognition in video. In: Huang, T.S., Sebe, N., Lew, M.S., et al. (Eds.), Computer Vision in Human-Computer Interaction: ECCV 2006 Workshop on HCI, Graz, Austria, May 13. Proceedings. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 64–77.

Divakaran, A., 2009. Multimedial Content Analysis: Theories and Applications. Springer.

Dragicevic, P., et al., 2008. Video browsing by direct manipulation. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, Florence, Italy, pp. 237–246.

Feng, J., et al., 2010. Computer usage by children with down syndrome: challenges and future research. ACM Transactions on Accessible Computing 2 (3), 32.

Fouse, A., et al., 2011. ChronoViz: a system for supporting navigation of time-coded data. In: CHI'11 Extended Abstracts on Human Factors in Computing Systems. ACM, Vancouver, BC, Canada, pp. 299–304.

Glaser, B.G., 1978. Theoretical Sensitivity: Advances in the Methodology of Grounded Theory. Sociology Press, Mill Valley, CA.

Glaser, B.G., 1992. Emergence vs. Forcing Basics of Grounded Theory Analysis. Sociology Press, Mill Valley, CA.

Glaser, B.G., Strauss, A.L., 1967. The Discovery of Grounded Theory: Strategies for Qualitative Research. Aldine, Chicago.
Guo, H., Wang, J., Xu, M., Zha, Z., Lu, H., 2015. Learning multi-view deep features for small object retrieval in surveillance scenarios. In: Proceedings of the 23rd ACM International Conference on Multimedia, pp. 859–862.
Hallgren, K.A., 2012. Computing inter-rater reliability for observational data: an overview and tutorial. Tutorial in Quantitative Methods for Psychology 8 (1), 23–34.
Hanjalic, A., et al., 2006. Multimedia content analysis, management and retrieval. In: Proceedings of the International Society for Optical Engineering (SPIE), pp. 6073.
Holsti, R., 1969. Content Analysis for the Social Sciences and Humanities. Addison-Wesley, Reading, MA.
Kandel, S., et al., 2008. Photospread: a spreadsheet for managing photos. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, Florence, Italy, pp. 1749–1758.
Krippendorff, K., 1980. Content Analysis: An Introduction to its Methodology. Sage Publications, Newbury Park, CA.
Krippendorff, K., 2004. Reliability in content analysis. Human Communication Research 30 (3), 411–433.
Kuziemsky, C.E., et al., 2007. A grounded theory guided approach to palliative care systems design. International Journal of Medical Informatics 76, S141–S148.
Landis, J.R., Koch, G.G., 1977. The measurement of observer agreement for categorical data. Biometrics 33 (1), 159–174.
Li, Y., et al., 2006. Robust head tracking with particles based on multiple cues fusion. In: Huang, T.S., Sebe, N., Lew, M.S., et al. (Eds.), Computer Vision in Human-Computer Interaction: ECCV 2006 Workshop on HCI, Graz, Austria, May 13. Proceedings. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 29–39.
Lyberg, L., Kasprzyk, D., 1997. Some aspects of post-survey processing. In: Lyberg, L., Blemer, P., Collins, M., et al. (Eds.), Suvery Measurement and Process Quality. John Wiley and Sons, New York.
Myers, M.D., 1997. Qualitative research in information systems. MIS Quarterly 21 (2), 241–242.
Myers, M.D., 2013. Qualitative Research in Business and Management, second ed. Sage Publications, Los Angeles, CA.
Neuendorf, K., 2002. The Content Analysis Guidebook. Sage Publications, Thousand Oaks, CA.
Norman, D., 2013. The Design of Everyday Things: Revised and Expanded Edition. Basic Books, New York, NY.
Norman, K.L., 1991. Models of the mind and machine: information flow and control between humans and computers. In: Marshall, C.Y. (Ed.), Advances in Computers. vol. 32. Elsevier, pp. 201–254.
Ohm, J., 2016. Multimedia Content Analysis. Springer.
Osgood, E.E., 1959. The representational model and relevant research. Trends in Content Analysis. I. de Sola Pool. University of Illinois Press, Urbana, IL.
Peltonen, P., et al., 2008. It's Mine, Don't Touch!: interactions at a large multi-touch display in a city centre. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, Florence, Italy, pp. 1285–1294.
Preece, J., et al., 2015. Interaction Design: Beyond Human-Computer Interaction. John Wiley & Sons, Ltd., West Sussex, UK.

Putnam, C., Chong, L., 2008. Software and technologies designed for people with autism: what do users want? In: Proceedings of the 10th International ACM SIGACCESS Conference on Computers and Accessibility. ACM, Halifax, Nova Scotia, Canada, pp. 3–10.

Rui, Y., Qi, G.-J., 2007. Learning concepts by modeling relationships. In: Sebe, N., Liu, Y., Zhuang, Y., Huang, T.S., et al. (Eds.), Multimedia Content Analysis and Mining: International Workshop, MCAM 2007, Weihai, China, June 30–July 1. Proceedings. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 5–13.

Saldaña, J., 2012. The Coding Manual for Qualitative Researchers. SAGE Publication, Thousand Oaks, CA.

Sears, A., et al., 2001. Productivity, satisfaction, and interaction strategies of individuals with spinal cord injuries and traditional users interacting with speech recognition software. Universal Access in the Information Society 1 (1), 4–15.

Sebe, N., et al., (Eds.), 2007. Multimedia content analysis and mining. In: Proceedings of the Multimedia Content Analysis and Mining International Workshop MCAM 2007. Lecture Notes in Computer Science.

Stake, R.E., 1995. The Art of Case Study Research. Sage Publications, Thousand Oaks, CA.

Stemler, S., 2001. An overview of content analysis. Practical Assessment, Research & Evaluation 7 (17), 137–146.

Strauss, A.L., 1987. Qualitative Analysis for Social Scientists. Cambridge University Press, Cambridge.

Suh, H., et al., 2016. Developing and validating the user burden scale: a tool for assessing user burden in computing systems. In: The ACM Conference on Human Factors in Computing Systems (CHI). ACM, San Jose, pp. 3988–3999.

Vondrak, M., Sigal, L., Hodgins, J., Jenkins, O., 2012. Video-based 3D motion capture through biped control. ACM Transactions on Graphics (TOG) – Proceedings of ACM SIGGRAPH. 31 (4), Article 27.

Weber, R.P., 1990. Basics of Qualitative Research: Grounded Theory Procedures and Techniques. Sage Publications, Newbury Park, CA.

Wilhelm, A., et al., 2004. Photo annotation on a camera phone. In: CHI'04 Extended Abstracts on Human Factors in Computing Systems. ACM, Vienna, Austria, pp. 1403–1406.

Wolf, K., Abdelrahman, Y., Landwehr, M., Ward, G., Schmidt, A., 2016. How to browse through my large video data: face recognition & prioritizing for lifelog video. In: Proceedings of the 15th International Conference on Mobile and Ubiquitous Multimedia, pp. 169–173.

Wrench, J., et al., 2013. Quantitative Research Methods. Oxford University Press, New York.

Wu, Q., et al., 2006. EigenExpress approach in recognition of facial expression using GPU. In: Huang, T.S., Sebe, N., Lew, M.S., et al. (Eds.), Computer Vision in Human-Computer Interaction: ECCV 2006 Workshop on HCI, Graz, Austria, May 13. Proceedings. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 12–21.

Xue, W., et al., 2006. Voice activity detection using wavelet-based multiresolution spectrum and support vector machines and audio mixing algorithm. In: Huang, T.S., Sebe, N., Lew, M.S., et al. (Eds.), Computer Vision in Human-Computer Interaction: ECCV 2006 Workshop on HCI, Graz, Austria, May 13. Proceedings. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 78–88.

Yin, R.K., 2014. Case Study Research: Design and Methods, fifth ed. Sage, Thousand Oaks, CA.

Zhu, G., et al., 2006. Action recognition in broadcast tennis video using optical flow and support vector machine. In: Huang, T.S., Sebe, N., Lew, M.S., et al. (Eds.), Computer Vision in Human-Computer Interaction: ECCV 2006 Workshop on HCI, Graz, Austria, May 13. Proceedings. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 89–98.