Chapter 1. An Introduction to Coding Streams of Language

In this chapter, we provide you with an introduction to coding streams of language. Beginning with a rationale for coding language, we also detail our commitments on several methodological issues. We then explain how to use this book, inviting you to adapt it in whole or in part to develop an appropriate analytic workflow, to choose your tools, and to follow its procedures. We close by articulating our aspirations, the challenges we have tried to address, and the sometimes technical quandries on which we have tried to provide some guidance. For those readers familiar with the 2004 *Analyzing Streams of Language*, we have also included a list of what is new.

Some Preliminaries

What Coding Is

Coding is the analytic task of placing non-numeric data into descriptive categories, assigning them to *codes*. The data that we will be concerned with coding in this book is *verbal data*, data in the form of words that usually combine to make up what we like to call a *stream of language*, a stream that we as readers or writers, listeners or speakers experience as a flow over time. When we code verbal data, we analyze this flow, breaking it up into a categorical array,

using a set of codes. We do this analysis to answer research questions, to better understand what the language is saying, doing, or revealing about the participants or about the situation in which the language has been used.

Any kind of verbal data can be coded. Varying in length, verbal data include the single word responses participants give in questionnaires, the quick posts that participants make in response to news articles, the full texts published in books, articles, and essays—and anything in between. Verbal data may come from conversations that need to be transcribed in order to be analyzed. Or they may come in print form, which may need to be scanned and converted using optical character recognition (OCR). And, increasingly, verbal data come in digital form, harvested from the web, sent in tweets, or published in digital databases. In most of these cases, verbal data are copious; words come fast and cheap in many contexts. They tell us a lot about what is going on, but we need to work to understand their underlying patterns. This is the work of coding streams of language.

Usually when we refer to coding, we are referring to an analytic process guided by a set of procedures—a procedural coding scheme—that tells the analyst how to categorize a segment of verbal data by defining and illustrating the use of each coding category. This is the primary kind of coding we deal with in this book. But we will also introduce readers to two other kinds of coding: automated coding, which uses digital searches to automatically identify members of a coding category, and enumerative coding schemes, which list all of the members of its coding categories. As we shall see in Chapter 4, these three kinds of coding can be used on their own or in combination.

Methodological Approaches to Verbal Data Analysis

Because verbal data are so ubiquitous, many different methodological approaches have been developed to deal with them. Figure 1.1 shows one attempt at displaying complex relationships among these approaches. While coding is an analytic technique used in many fields, it has primarily been developed in the field of communication studies under the term *content analysis* and in the social sciences, more broadly, under the term *qualitative research*.

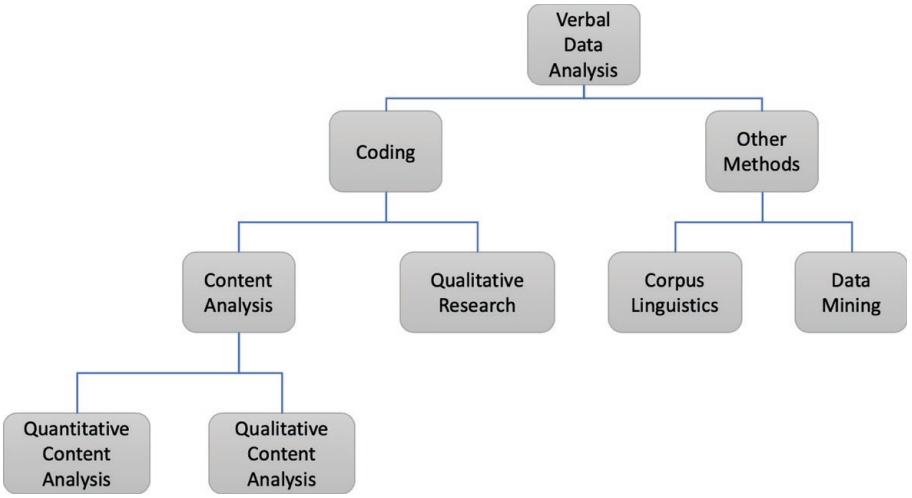

Figure 1.1: Taxonomy of approaches to verbal data analysis.

Traditional quantitative content analysis attempts to remove interpretation from coding. Often used for studies of media coverage, it provides coders with procedures using exact word matches or unambiguous judgments and uses quantification to look at overall patterns. By contrast, qualitative researchers, including those using qualitative content analysis, take an approach that is more interpretative. Many researchers adopt a qualitative approach as part of the process of choosing a CAQDAS (Computer-Aided Qualitative Data Analysis Software) tool such as Nvivo or Atlas.ti. Most though not all qualitative approaches to coding take a code as you go approach, and some, but not all, eschew any kind of quantification. In *Coding Streams of Language*, we take an interpretive approach to coding; that said, our commitment to being systematic and exploring patterns through numbers places us among the growing number of researchers taking a mixed-methods approach, which we discuss more fully in a later section.

Other methods for verbal data analysis exist that do not use coding. Approaches taken by corpus linguists, for example, focus on analyzing large sets of texts, often using some variety of grammatical or semantic tagging. In Chapter 2 on Designing the Analysis and in Chapter 4 on Coding Data, we

suggest ways that one kind of corpus tool, AntConc, can be used to explore and automatically code data.

Finally, emerging methods for data mining have been introduced to deal with large sets of verbal data. Using algorithmic rather than interpretive approaches, many big data approaches have little use for interpretation. But those who use machine learning methods to duplicate human judgment will often begin their work with the kind of coding we pursue.

The Important Role Coding Plays in Many Fields

We come to the coding of verbal data from the allied fields of writing studies and technical communication. No one should be surprised to find these language-intensive fields relying on a method that deals with verbal data. As we noted elsewhere (Geisler, 2018), coding is a key analytic method in writing studies and technical communication, being used in 44% of the research reports published in 2015 and 2016. These reports used a wide range of data. For example, Breuch et al. (2016) coded interview data from hospital patients and their families for recurring themes. Martinez et al. (2015) coded video data for the cognitive activities students used while writing syntheses.

Coding plays an important role in a far wider range of fields than this brief sample of studies might suggest. Any field that deals with humans as social beings, that collects naturally occurring language data or elicits such data from participants, will find a use for coding:

- In applied linguistics, Wyrley (2010) used coding to study communication practices in radiotherapy.
- In education, Stevenson (2013) used coding to study the linguistic strategies used by fifth grade bilingual students in science.
- In engineering education, Richter and Paretti (2009) used coding to analyze how engineering students reacted to multidisciplinary design.
- In information science, Nobarany and Booth (2014) used coding to examine the use of politeness strategies in open peer review.

- In human-computer interaction, Friess (2012), used coding to study the use of personas in software design.
- In legal studies, Jameson, Sohan, and Hodge (2014) used coding to better understand turning points in mediation.
- In environmental studies,Thompson (2005) used coding to examine the kinds of issues that were discussed in newspaper articles about a proposed off-shore wind power project.
- In public health, Banna et al. (2016) used coding to make a cross-culture comparison of ideas about healthy eating among Chinese and American undergraduate students.
- In operations management, Mugurusi and Bals (2016) use coding to study the stages of an offshoring strategy adopted by a purchasing and supply organization.

When to Code Verbal Data—Or Not

The coding we introduce in *Coding Streams of Language* is best used when three conditions hold:

1. You are looking for recurrent phenomena within and across streams of language,.
2. You are interested in understanding underlying patterns of doing and meaning in these streams.
3. You and your co-researchers have sufficient intuitions about these streams to place them into appropriate coding categories.

Let's take a look at these conditions one at a time.

First, coding is a procedure designed to detect recurrent patterns in a stream of language. If you are looking for phenomena that occurs rarely, the procedural coding we recommend in *Coding Streams of Language* would be more complex than the rewards would justify. For example, if you are looking for the turning point in a conversation, and you expect there to be, at most, one turning point or perhaps none at all in a given stream, you may be better

off using a careful close reading to find it. You might still find useful some of the techniques we describe in Chapter 4 for creating an explicit definition for yourself and your readers, but the other procedures described in this book would be more than you need.

Second, the analytic work we recommend in *Coding Streams of Language* is designed to examine the underlying patterns of meaning and doing, the ways with words of which participants may be largely be unaware. If, however, you are not concerned with the ways specific words and phrases are deployed and responded to, if you only want to identify the places in which certain topics are discussed, then you may only need to use a more simple topical coding (Geisler, 2018; Saldaña, 2016).

Finally, procedural coding, the primary method described in *Coding Streams of Language*, is designed to guide coders intuitions toward appropriate coding decisions. As we describe more full in Chapter 4, in some situations, no one outside of the context in which a stream was originally produced may have good intuitions about what the language means or how it works. The level of jargon and specialized knowledge may simply prevent outsiders from understanding what is going on from what is being said. If, for example, your verbal stream is in a language you do not understand, you obviously won't have the intuitions to code it.

But even if you fully understand the language of a verbal stream, you may not have the intuitions to code it appropriately. In this situation, you have two options. One option is to invite an informant, someone who is familiar with the context of production, to work with you as a coder. Another option is to use the enumerative coding, as described in Chapter 4, in which you list all of the possible words or phrases that you include within a coding category. An enumerative coding scheme has the benefit both of being transparent to your readers and of helping them to better understand intuitively what you intend.

To summarize, we invite you to use the procedures in *Coding Streams of Language* to code verbal data when you are looking for recurrent and underlying patterns in streams of language and about which you or your co-researchers have adequate intuitions.

The Patterns Revealed by Coding

As we discuss in this book, coding can be used to examine three basic kinds of patterns. The simplest pattern is the one-dimensional analysis we describe in Chapter 6, which asks how verbal data is distributed across a set of coding categories, often across a built-in contrast. Banna et al. (2016), for example, used a built-in contrast across Chinese and American undergraduates to notice differences in the ways they thought about healthy eating. Based on these distributional differences, Banna and colleagues recommended different public health strategies be used in these two communities.

Verbal data that have been coded with more than one coding scheme can be looked at multidimensionally, as we introduce in Chapter 7. Jameson et al. (2014), for example, analyzed conversational interactions that occurred during mediation along two distinct dimensions. First, they coded the precipitants leading to turning points in negotiations, points in which the relationship between the disputants seem to change. Second, they coded for negotiation outcome. This allowed Jameson and colleagues to look for relationships between the two dimensions, the kind of precipitants used, and the outcomes of the mediation. Based on the relationships they saw, they suggested ways that mediators could be more helpful.

The third pattern that can be revealed by coding is temporal. As we acknowledge in Chapter 8, temporal analysis deserves to be used more often for what it shows us about streams of language. Mugurusi and Bals (2016) use a kind of temporal analysis to show how the dimensions of Centralization, Participation, Formalization, Standardization, and Specialization changed over four phases in the offshoring process. The authors concluded that the offshoring process may be more disjointed and non-linear than current models in operations management would suggest.

Our Core Commitments

We bring to the task of coding streams of language a set of commitments that we'd like to put on the table from the start. They have served as our points of

departure for the process and procedures that you will find in the rest of the book. In this section, we make these commitments explicit not so much to argue for them but so that you can judge for yourself.

Commitment to Being Procedural

Coding Streams of Language is fundamentally a procedural guide. That is, it provides you with a set of step-by-step procedures for coding and then analyzing verbal data. We anticipate that, as you grow in experience, you will modify, extend, and even discard these procedures. But our intention is to provide you a very clear basis with which to begin.

You will find that most of this procedural knowledge has not been documented elsewhere. Instead, it most often handed down mentor to student during office hours or shared peer to peer in late night sessions. The trouble with these practices is that they tend to keep cultural knowledge about analysis within a closed inner circle. Not only does this seem unfair to us, but it also keeps these procedures out of the light of day. So we put our procedures out there for you to see, use, question, and refine.

Commitment to the Systematicity of Coding

Coding Streams of Language aims to help you produce a systematic analysis. To be systematic means to follow some articulate orderly procedure. It does not mean you have abandoned intuition—more about this later—but it does mean that you have tried as far as possible to create an analysis that can be replicated: that the coding decisions you make today will be the ones that you agree with tomorrow; that the coding decisions your co-researchers make will be more or less the ones that you would make.

The commitment to systematicity lies behind the importance we give to segmenting verbal data in advance of coding it. And, as we introduce in Chapter 3, choosing the right unit for segmentation is key to developing a coding scheme that works. The commitment to systematicity also lies behind our emphasis on reliability. In Chapter 5 we describe how having someone else try to

code your data and then comparing it to your own coding is the eye-opening key to developing a good coding scheme.

Commitment to the Design of Analysis

Coding Streams of Language urges you to design your analysis. Verbal data tends to pile up and overwhelm the best of us. Stepping back to consider how you will design your analysis can help you get a handle on what can otherwise be an enormous task.

In Chapter 2, we suggest that you begin with some initial explorations, sharpening your intuitions about what looks interesting. Then we give you some options on sampling your data, using your research questions to pick out a manageable subset of your data for further in-depth analysis. And finally, we recommend that you build your analysis around a built-in contrast, looking not only at data that you think should reveal the phenomenon in which you are interested, but also at data in which you expect the phenomenon to be absent. Sometimes the best way to know what you're looking for it to see its absence.

Commitment to the Complexity of Language

Coding Streams of Language takes a rhetorical approach to coding. That is, it acknowledges the complexity of language use. It considers not just what language says—that is, its topics—but also what language does. It assumes that language is more than just a vessel for content, more than a series of topics; that it *does* as well as *means*.

Acknowledging the complexity of language also requires us to forgo the expectation that any coding scheme can be absolutely unambiguous. Language will always require the interpretive powers of a language user. Coding does not replace the human coder but provides a guide to our intuitions. The role that context plays in developing these intuitions is inescapable. What words and phrases mean in one context might be quite different in another context. Coding depends, however, on the idea that these intuitions can be developed using a full coding scheme as we discuss in Chapter 4.

A Commitment to Mixed Methods

In *Coding Streams of Language*, we take a mixed-methods approach to the analysis of verbal data. Adapting the terminology introduced by Vogt et al. (2014), the workflow we advocate moves from coding in words to an analysis that combines qualitative (words), quantitative (numbers), and graphic (charts) representations. Like many mixed-methods researchers, we no longer find it useful to see qualitative and quantitative approaches as opposing methodologies, but rather prefer to see them as constituting a useful set of tools (Sandelowski et al., 2009).

Nevertheless, our commitment to mixed methods has lead us to adopt the standard of mutual exclusivity for coding. Mutual exclusivity refers to the requirement that each segment of data should be assigned to one and only one code. Mutual exclusivity is often seen as one of the major dividing practices between qualitative and quantitative approaches to coding. Examined more closely, however, we have found that these two analytic traditions are often closer than we might expect because language is inherently multidimensional.

In practical terms, multidimensionality often means that an analyst considering how to code a piece of language often sees multiple ways to code it. This will be true whether one is approaching coding from the perspective of content analysis, in which the goal is to create mutually exclusive categories, or from the perspective of qualitative analysis, in which double coding is not uncommon. Our method for dealing with the tendency to double code is to dimensionalize the data. As we describe in Chapter 4, rather than seeing the inclination to double code as arising from irreconcilable options, we can turn it into an invitation to develop mutually exclusive codes in different dimensions.

Our commitment to mixed methods also keeps us open with respect to research designs. We agree with Vogt et al. (2014), that the choice of analytic methods is not predetermined by the design of your study. Whether you have collected data in the context of a tightly-controlled experimental investigation or as a result of an extended stay in the field, as long as you have verbal data, you can code it and analyze it following the procedures we lay out in this book.

Using This Book

We have organized the material in this book to support your coding work in three distinct ways: first, with a distinctive workflow, second, with a distinctive set of tools, and finally, with a distinctive set of procedures. We briefly introduce each of these below.

The Workflow

In keeping with its procedural nature, *Coding Streams of Language* is organized around a workflow that you can adopt in part or whole for your analytic endeavors. A bird's eye view is shown in Figure 1.2. The first five components, shown in green in the figure, take you through the heart of coding and conclude with your having a set of data coded with a reliable coding scheme. The next three components, shown in blue, provided techniques for visualizing the patterns revealed by this coding. And the final two components, shown in orange, include ways to check the significance of those patterns and detail their results for readers.

We have made this book available in whole or by chapter to allow you to adopt this entire workflow or to pick and choose depending on your needs, interests and the state of your investigation. Below, we describe what is covered in each chapter so that you may target your reading.

Chapter 2: Designing the Analysis

If you are just beginning your project, with some data in mind but not yet collected, or with some data collected but not yet analyzed, you may want to start with Designing the Analysis in Chapter 2. Chapter 2 will suggest ways to focus on a specific phenomenon, articulate research questions, and develop a strategy for sampling from what may be a large universe of potential data.

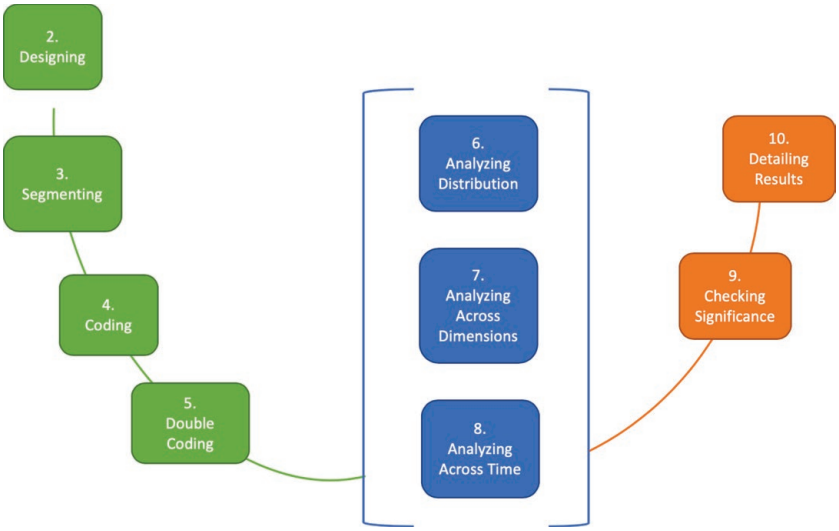

Figure 1.2: The workflow in Coding Streams of Language.

Chapter 3: Segmenting

If you have data in hand and some idea of what you are interested in, start with *Segmenting* in Chapter 3. A necessary precursor to coding, segmenting involves dividing your stream of language into units appropriate to the phenomenon in which you are interested. In this chapter, we provide a range of options for segmenting not discussed elsewhere in the literature, including basic grammatical units common to all language, more specific units characteristic of conversation and written texts, and interesting linguistic features such as indexicals, personal pronouns, modals, and metadiscourse.

Chapter 4: Coding the Data

Once you have segmented data in hand, you will want to turn to *Coding the Data* in Chapter 4. There you will learn about the components of a full procedural coding scheme and be guided through the iterative process of building one. Chapter 4 also introduces techniques for automated and enumerative coding as well as ways these can be combined.

Chapter 5: Achieving Reliability

If you have already coded your data, you may want to consider Chapter 5 on *Achieving Reliability*. To insure that you have a coding scheme that makes sense and is consistent, Chapter 5 suggests you invite a second coder to code using your coding scheme, check for intercoder agreement, and then use the results to improve your coding scheme. By the end of this chapter, you should have a set of data coded with a reliable coding scheme.

Chapter 6: Seeing Patterns of Distribution

Chapter 6 is the first of three dealing with seeing patterns in data that have been fully coded. Include *Seeing Patterns of Distribution* in your workflow in order to detect patterns in the way that your data is distributed among the categories of a single coding scheme. In the process, you will learn how to build a frequency table and distribution graphs like those shown in Figure 1.3.

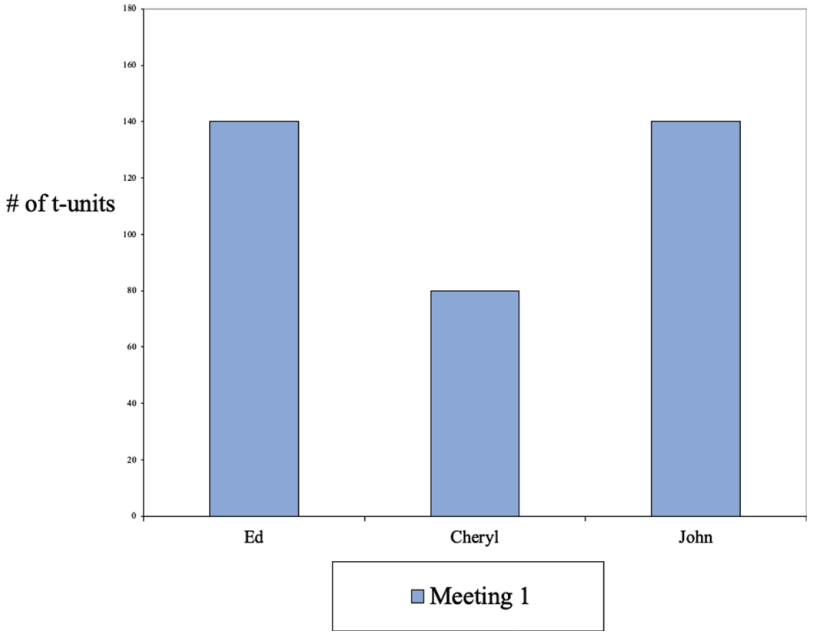

Figure 1.3: A sample distribution graph.

Chapter 7: Exploring Patterns Across Dimensions

If you have coded your data using more than one coding scheme, you may want to explore the relationships among the schemes. How does the pattern revealed in one dimension relate to patterns in a second dimension? Chapter 7, *Exploring Patterns Across Dimensions*, walks you through building contingency tables and block charts like that shown in Figure 1.4, as well as making stepwise comparisons of the dimensional patterns to check their relationships.

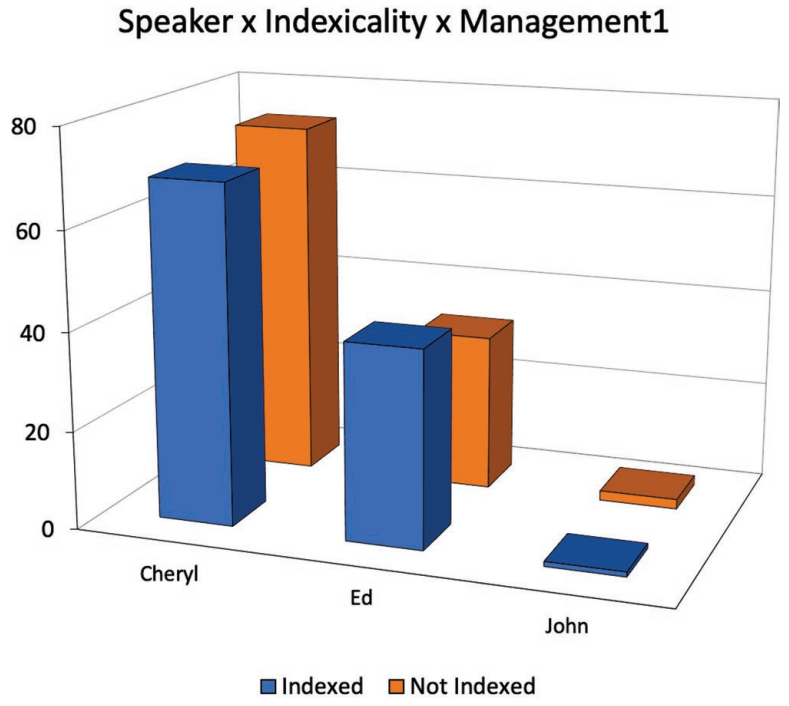

Figure 1.4: A sample block chart.

Chapter 8: Following Patterns over Time

If you want to understand the way that a stream of language unfolds over time or to compare two or more different streams of language, you may find Chapter 8 on temporal analysis useful. Coded verbal data that has been taken from

intact cultural artifacts—complete texts, full conversations, an extended thread, and so on—often exhibit a distinctive temporal shape like the one shown in Figure 1.5. *Following Patterns over Time* will help you to uncover and understand these temporal shapes.

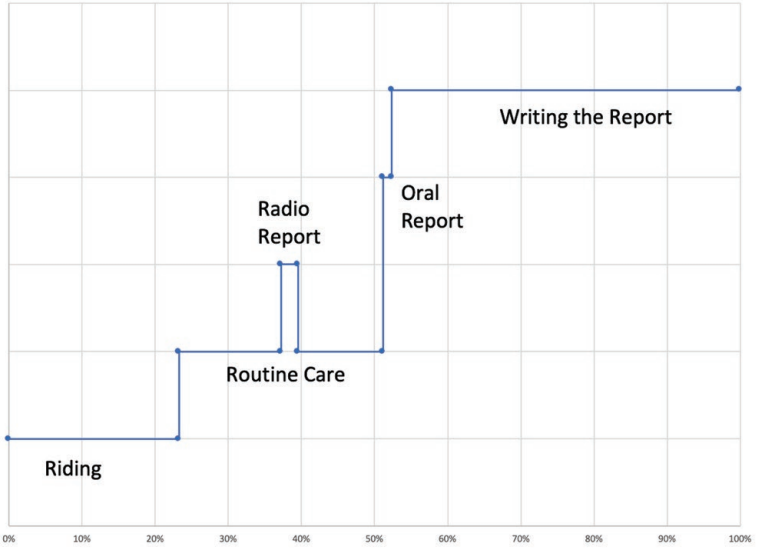

Figure 1.5: A sample temporal index showing the use of language in an ambulance run.[1]

Chapter 9: Evaluating Significance

If you have noticed interesting differences in the distribution of coding categories across your data, you may want to better assess their significance. Are these differences big enough to matter? *Evaluating Significance* shows you how to compare the actual distribution of your data across coding categories with the distribution that would be expected if these differences were not significant. Not all researchers will want or need to measure significance, but Chapter 9 provides some techniques for those who do.

1 Adapted from Geisler & Munger (2001).

Chapter 10: Detailing the Analysis

If you have coded your data and analyzed its patterns, you will want to find the best way to communicate the results to your readers. In Chapter 10, we show you how to make the link between the overall patterns you have uncovered and the details of language use. Use *Detailing the Analysis* to make the language patterns come alive for your readers.

The Tools

In 2004, Geisler published *Analyzing Streams of Language* using procedures in Microsoft Excel. For *Coding Streams of Language*, we have made significant updates to these procedures as well as added a second set of procedures using MAXQDA, a Computer Assisted Qualitative Data Analysis Software (CAQDAS) package. We have also incorporated some useful procedures using Microsoft Word and AntConc, a concordance program. We do not assume any working knowledge of these software packages on our readers' part in advance of trying out the procedures, although we do recommend using a free trial to get comfortable.

Microsoft Excel is the traditional tool used for coding verbal data (Geisler, 2004). Although most people think of it as a quantitative tool, Excel can function as a database manager, functionality that is important to coding. Many academics have access to Excel through their university's educational program for Microsoft Office, but if one were to buy an individual educational license to use it off-line for a year, the cost would be around $150 US. It runs on both Windows and Mac platforms,[2] and you can use it to accomplish all of the work in *Coding Streams of Language*.

MAXQDA, our second major tool, is one of a growing class of CAQDAS tools that support mixed-methods analysis. Developed in 1989 by Udo Kuckartz to deal with political discourse, it is supported by VERBI GmbH out of Berlin, Germany. It is one of the oldest CAQDAS programs and has a strong reputation as an efficient and responsive tool (Schmeider, 2014; Silver & Lew-

[2] Readers who prefer to use Google sheets will find that most of the procedures in this book appear to be adaptable to this tool, although we have not tested this directly.

ins, 2017). It can also be expensive unless you're a student. A single license for a regular user is over $500 USD, but students can get access for two years for under $100. We particularly like MAXQDA for its ease of use, use of color, and easy integration with Excel. It does not do everything we include in *Coding Streams of Language*, but it does many things superbly. And it is relatively easy, when necessary, to move into Excel to complete tasks. Our procedures provide you with explicit directions for managing this integration.

In *Coding Streams of Language*, we have created procedures that stay faithful to our commitments to approach coding systematically and with respect for language complexity. Neither Excel nor MAXQDA were designed for this task. Each is a general purpose tool that we have adapted to our needs, sometimes easily and sometimes by using complex workarounds (see Geisler, 2018). We invite you to extend, modify, and reinvent our procedures and, by all means, share them with us at https://wac.colostate.edu/books/practice/codingstreams/.

The Procedures

Procedures have been formatted to facilitate their use. Excel procedures always appear first with the parallel MAXQDA procedure(s) following. Other procedures in tools like Microsoft Word and AntConc will be found interspersed throughout the chapters.

Procedures appear in a distinctive typeface with numbered steps, are numbered and labeled, and have their own table of contents at the front of the book. All of these formatting conventions are designed to make them easy to find and follow. We often find ourselves paging to a specific procedure to remember "how do I do that?" We anticipate you will too.

Screencasts on YouTube

Every procedure in this book has a screencast video where you can watch the procedure in action. The URL following the procedure name is a direct link to the appropriate YouTube video playlist. If you are reading a chapter electronically, click on the link to access the chapter playlist and then choose the

video from that list. If you are using this book in print, you can go to our website at https://wac.colostate.edu/books/practice/codingstreams/ for the links or go to our YouTube channel at https://www.youtube.com/channel/UCYi7qEAnSOvkMzCMogMkW5g/playlists.

Technical Questions and Where to Find the Answers

The following list includes technical questions that many readers are searching for. It is both more fine-grained and less comprehensive than the chapter-by-chapter summary we provided earlier. If you're looking for something specific and slightly complicated, we expect you will be able to find it here.

How do I decide what to code?	Chapter 2
How do I choose the right unit to segment my data?	Chapter 3
How can I easily number my segments?	Excel Procedure 3.1 & 3.2
How can I automatically code my data?	Excel Procedure 4.7 & MAXQDA Procedure 4.8
How much of my data do I have to double code?	Chapter 5
How much agreement is enough agreement?	Chapter 5
Which reliability test should I use?	Chapter 5
How do I check intercoder agreement?	Excel Procedure 5.3 & MAXQDA Procedure 5.3
How do I calculate interrater reliability?	Excel Procedure 5.7, Excel Procedure 5.1, MAXQDA Procedure 5.7, & Procedure 5.1
How do I make a frequency table?	Excel Procedure 6.2 & MAXQDA Procedure 6.1
How do I make a pivot table?	Excel Procedure 5.5

How do I make a distribution graph for a single coding dimension?	Excel Procedure 6.3 and MAXQDA Procedure 6.2
How do I make a contingency table?	Excel Procedures 7.1, 7.2, and 7.3
How do I make a block chart to see the relationships across two dimensions?	Excel Procedure 7.4
How do I make a temporal graph to see distribution over time?	Excel Procedure 8.2 and MAXQDA Procedure 8.2
How do I calculate subtotals for aggregates?	Excel Procedure 8.6
Is a chi-square test enough?	Chapter 9
How do I calculate chi-square?	Chapter 9

What's New since 2004 *Analyzing Streams of Language*

For readers who are using *Analyzing Streams of Language*, here is what is new in *Coding Streams of Language*:

- All procedures in Excel have been updated. Data summarizing the use of pivot tables and countifs functions have been included.
- Procedures for using Computer-Aided Qualitative Data Analysis Software (CAQDAS) have been included using MAXQDA.
- Screencasts have been created for all procedures and are available at our YouTube channel at https://www.youtube.com/channel/UCY-i7qEAnSOvkMzCMogMkW5g/playlists.
- Procedures are more explicit and easier to find.
- Memoing as a way to reflect and document your analytic process has been included.
- The use of a concordance program, AntConc, for data exploration is explained.
- Conceptual discussions of major issues in coding open each chapter.

Our Aspirations

Our hope for this volume in its entirety and in each chapter individually is to guide you through the process of doing verbal data analysis. In the end, we would like each reader to reach a point of comfort and proficiency with the concepts and techniques outlined in this book to conduct good research and to pass along those skills and perspectives to others.

One way that we aim to address this grand ambition is to provide you with a method for doing verbal data analysis that has plenty of conceptual and procedural scaffolding. We do not assume that you have knowledge of coding and analysis or that you are beginning from anywhere other than square one. Our aim is to help you develop your own analysis, develop a robust and reliable coding scheme, analyze coding patterns in your data, and then link those patterns back to the scholarly or professional conversations you participate in.

A related aim is to provide you with a comprehensive and systematic approach to studying verbal data analysis that makes its analytic assumptions plain. We invite you to reflect on those assumptions and recognize the way that the analytic techniques we describe derive from those assumptions. As your needs warrant, we want you to reflect on your own assumptions and then adapt and extend the techniques to your research situation. The procedures discussed in this book are extensible.

Above all, our aim is to make this method of verbal data analysis accessible to a broad range of scholars in rhetoric and writing studies while at the same time reaching out to scholars in other fields who may be grappling with streams of verbal data without a clear notion of how to derive meaning from those streams in a way that is disciplined, systematic, and reliable.

Selected Studies Using Coding

Banna, J. C., Gililand, B., Keefe, M., & Zheng, D. (2016). Cross-cultural comparison of perspectives on healthy eating among Chinese and American undergraduate students. *BMC Public Health*. Retrieved from https://bmcpublichealth.biomedcentral.com/articles/10.1186/s12889-016-3680-y

Breuch, L. K., Bakke, A., Thomas-Pollei, K., Mackey, L. E., & Weinert, C. (2016). Toward audience involvement: Extending audiences of written physician notes in a hospital setting. *Written Communication*, *33(4)*, 418-451.

Friess, E. (2012). Personas and decision making in the design process: An ethnographic case study. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, *USA,* 1209-1218.

Jameson, J. K., Sohan, D., & Hidge, J. (2014). Turning points and conflict transformation in mediation. *Negotiation Journal*, *30(2),* 209-229.

Martínez, I., Mateos, M., Martín, E., & Rijlaarsdam, G. (2015). Learning history by composing synthesis texts: Effects of an instructional programme on learning, reading and writing processes, and text quality. *Journal of Writing Research, 7(2),* 275-302.

Mugurisi, G., & Bals, L. (2017). A processual analysis of the purchasing and supply organization in transition: The impact of offshoring. *Operations Management Research*, *10(1-2),* 64-83.

Ngai, C. S. B., & Jin., Y. (2016). The effectiveness of crisis communication strategies on Sina Weibo in relation to Chinese publics' acceptance of these strategies. *Journal of Business and Technical Communication*, *30(4),* 451-494.

Nobarany, S., & Booth, K. S. (2015). Use of politeness strategies in signed open peer review. *Journal of the Association for Information Science and Technology, 66*(5), 1048-1064.

Stevenson, A. (2013). How fifth grade Latino/a bilingual students use their linguistic resources in the classroom and laboratory during science instruction. *Cultural Studies of Science Education*, *8(4),* 973-989.

Thompson, R. (2005). Reporting offshore wind power: Are newspapers facilitating informed debate? *Coastal Management*, *33(3),* 242-262.

Wyrley, B. (2010). "Talking technical": Learning how to communicate as a health care professional. *South African Linguistics and Applied Language Studies, 28*(3), 209-218.

Richter, D. M., & Paretti, M. C. (2009). Identifying barriers to and outcomes of interdisciplinarity in the engineering classroom, *European Journal of Engineering Education*, *34(1),* 29-45.

For Further Reading

Geisler, C. (2004). *Analyzing streams of language: Twelve steps to the systematic coding of text, talk and other verbal data*. London: Pearson/Longman.

Geisler, C. (2018). Coding for language complexity: The interplay among methodological commitments, tools, and workflow in writing research. *Written Communication 35*(2), 215-249.

Saldaña, J. (2016). *The coding manual for qualitative researchers* (3rd ed.). Los Angeles: Sage.

Sandelowski, M., Voils, C. I., & Knafl, G. (2009). On quantitizing. *Journal of Mixed Method Research 3*(3), 208-222.

Shuy, R. (2018). What are "allness terms"? *The Chronicle of Higher Education*. Retrieved from https://www.chronicle.com/blogs/linguafranca/2018/10/08/what-are-allness-terms/

Vogt, W. P., Vogt, E. R., Gardner, D. C., & Haeffele, L. M. (2014). *Selecting the right analyses for your data: Quantitative, qualitative, and mixed methods.* New York: Guilford Press.