Chapter 4. Coding the Data

> In this chapter, you will code the data you segmented in the previous chapter. After you devise an initial start list of codes, you will use an iterative process to move back and forth between sample data and coding scheme to develop a procedural coding scheme that best tracks the phenomenon that interests you. You will also be introduced to automated coding and to using an enumerative coding scheme.

Concepts in Coding

Coding is the analytic task of assigning codes to non-numeric data. Coding language data is a technique used in a variety of research traditions. In traditional content analysis, coding falls under the heading of "human coding" and makes use of a codebook which, according to Neuendorf (2016) should be set up in advance of coding and should be "so complete and unambiguous as to almost eliminate the individual differences among coders" (Chapter 5, Section on Codebooks and Coding Forms, para. 1). In qualitative analysis, coding is treated as an activity that creates and assigns a word or phrase to symbolize, summarize, or otherwise capture some attribute of "a portion of language-based or visual data," often in interaction with that data (Saldaña, 2016, Chapter 1, Section on What is a Code?, para. 1). Finally, in text mining, especially that using supervised machine learning, language data is often coded in a first stage of work to create a corpus from which a machine can then "learn" (Geisler, 2016b; Omizo & Hart-Davidson, 2016a).

Coding Schemes

The way that the segments are assigned codes is governed by some kind of coding guide. In content analysis, guides to coding are gathered together into a *codebook*, which provides analysts with detailed directions for coding along all the dimensions defined for the data. A draft codebook is developed based on the literature and then further refined during coder training. It reaches its final form before final coding. Neuendorf (2016) provides an example of a codebook for female characters in James Bond films (Box 5.1) that provides directions for coding such simple dimensions as Name ("If the character's name is provided, list this name . . .") and more complex dimensions such as this one for Physical Appearance:[4]

> Indicate whether the female character is extremely attractive (professional model status), attractive (very pleasant looking, above average), average (pleasant looking, but average in looks), below average (not pleasant looking, below average looks), extremely unattractive (extremely unpleasant looking, considered homely). (Chapter 5, Box 5.2, para. 14).

In qualitative analysis, codes may be gathered together into a coding system (MAXQDA), a coding tree (Dedoose) or, more generically, a coding scheme (Silver & Lewins, 2017). These coding schemes may be developed in advance of the coding, but undergo substantial refinement throughout the coding process. Particularly during second cycle coding, groups of codes may be collected together, rearranged, subsumed one under another, and so on. One example of an emergent category cited by Saldaña (2016) from Maykut and Morehouse (1994) provides a rule for inclusion along with sample data:

> **Physical Health:** The participant shares matters related to physical health such as wellness, medication, pain, etc.: "I'm on 25 milligrams of amitriptyline each night"; "I've lost ten pounds on this new diet." (p. 11)

[4] We might cringe at the subject, but this is one of only a few clear, accessible examples of a complex coding scheme.

For our purposes, a coding scheme should articulate clearly the procedures used to code your data. It serves two functions. In the next stage of analysis, it will serve as a set of directions to a second coder. Later on, when you publish your study, it will serve to let your reader know how you have defined your categories and how you have assigned data to them.

A sample coding scheme can be found in Figure 4.1. As you can see, it is written as a set of directions to a coder and includes the following:

- The name of the dimension being coded for. In Figure 4.1, the name of the dimension being coded for is the World of Discourse. The name of the dimension should be clearly identified at the top of the coding scheme.
- The names of the coding categories. In Figure 4.1, these include *Rhetorical Process*, *Domain Content*, and *Narrated Cases*, and are clearly labeled in bold type.
- The unit of analysis to which the codes are to be applied. In Figure 4.1, the unit mentioned as being coded is the t-unit.
- A definition for each coding category. In Figure 4.1, the definition for *Rhetorical Process*, for example, is given as "the worlds in which people make claims as authors."
- An enumeration of the kinds of cases that the coding category includes. In Figure 4.1, *Rhetorical Process* includes cases referring to texts as well as cases referring to authors.
- One or more examples of each case. Examples of authors in Figure 4.1 include "Childress," "I," and "you."

Coding the World of Discourse

The following figure shows an example of a procedural coding scheme.

Code as **Rhetorical Process** any t-unit that refers to the worlds in which people make claims as authors. This includes referring to:

- the texts, or parts of texts, in which claims are made: "the book," "the introduction";

- the authors of claims, including the students and teachers as makers of claims: “Childress,” “I,” “you”;
- nominals that characterize actions taken by authors as claim makers: “question-begging”;
- descriptions of the interactions among authors as claim makers: “discussions,” “disagreements”;
- the requirements or directions for the assignment: “a paper in two sections”;
- general categories of claims that can be made by authors: “a definition,” “a justification,” “a reason,” “a question”; and/or
- any consideration or product relevant to the generation of claims: “my notes.”

Code as **Domain Content** any t-unit that refers to the world of truths about paternalism. These truths have an external flavor to them: They are said to be true independent of any claim. This will include t-units referring to:

- philosophical concepts related to paternalism: “paternalism,” “respect for persons”;
- terms for the standard components of paternalism taken in an abstract way: “agent,” “actions,” “features,” “case”;
- relationships between these concepts or components: “connection”; and/or
- characteristics, either positive or negative, of these concepts or components: “principle.”

Code as **Narrated Case** any t-unit that refers to particular worlds in which (paternalistic) narratives take place that are taken to exist independently of domain concepts. These will include:

- specific people or actions that are taken to exist independently of the concepts in the domain but that may potentially be characterized with respect to these concepts;
- general categories of people or actions that are taken to exist independently of the concepts in the domain but that may potentially be characterized with respect to these concepts;

- characteristics of specific or general people or actions that are taken to exist independently of the concepts in the domain but that may potentially be characterized with respect to these concepts; and/or
- "you" or "I" when cast in a role involving an action that is taken to exist independently of the concepts in the domain but that may potentially be characterized with respect to these concepts.

Figure 4.1: Sample procedural coding scheme.

The ultimate goal of coding verbal data is to label segments of verbal data in a way that represents the phenomenon in which you are interested and to do so systematically and reliably. Keep in mind that due to the complexity of language, coders will always need to draw on their intuitions about what language does and means. No amount of effort in constructing a coding scheme will eliminate the need for a coder to use interpretive judgment in coding. It is impossible, in other words, for a coding scheme to be as "complete and unambiguous" as Neuendorf (2016) exhorts. Instead, the best coding schemes work with, massage, and otherwise direct a coder's intuitions into ways of interpretation intended by the researcher.

Mutual Exclusivity in Coding

Although there are variations that we discuss at the end of this chapter, in general, each segment should be assigned to one and only one code. That is, the codes should be mutually exclusive. Mutual exclusivity refers to the requirement that each piece of data should be assigned to one and only one code. It is often seen as one of the major dividing practices between qualitative and quantitative approaches to coding. In fact, some content analysts suggest that non-mutual exclusivity is a "fatal flaw" in qualitative approaches (Boettger & Palmer, 2010; Bourque, 2004; Stemler, 2001). Examined more closely, however, these two analytic traditions are often closer than we might expect. In fact, mutual exclusivity is best understood in terms of what we'll call the dimensionality of verbal data.

Language is inherently multidimensional. As a consequence, coding frequently has dimensions to it as well. In practical terms, this 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.

Where content coding and qualitative coding do differ is in what they do in the face of this inclination to double code. Krippendorff (2013), one of the leading scholars of content analysis, suggests that the inclination to double code arises out of a conceptual confusion over what one is looking for (p 132). In this case, the best solution is to more clearly distinguish among the dimensions of the data. In this way, a scheme that looks like it is not mutually exclusive might be developed into a set of dimensions.

For example, in developing a coding scheme for a set of interviews with students from a writing class, an analyst might first be inclined to create a four-code system.[5]

> **Concept:** Code as Concept any explanation of a concept covered in class.
>
> **Process:** Code as Process all expressions of the student engaging in any process related to writing.
>
> **Value:** Code as Value any expression of beliefs or values about the concept or process.
>
> **Problem:** Code as Problem, any instance of expressing a problem with a process.

The analyst finds, however, that she is inclined to double code statements like the following:

> "I'm a real procrastinator, so I just start writing without an outline."

Here the analyst sees evidence of a problem (*procrastinator*), a process (*start writing*), and a concept (*outline*).

[5] We want to thank Barbara Bird, Dean of Faculty Development at Taylor University, for allowing us to share this example with readers.

Following Krippendorf, we find opportunities to untangle multiple dimensions in this scheme. Even the code definitions themselves seem to suggest one dimension related to writing topics (concepts or processes), and perhaps two other dimensions, one related to values and the other related to problems (although these two might be able to be combined). The Writing Topics dimension would have two mutually exclusive codes (concept or process). The statements in the Values and Problem dimensions might just be coded as present or absent, or might be further developed to reflect possible kinds of values (positive, negative . . .) or problems.

Overall then, a one-dimension scheme that originally contained four categories that were not mutually exclusive could be developed into a three-dimensional analysis where each dimension had its own mutually exclusive codes. Thus, as Neuendorf (2016) seems to suggest, mutual exclusivity can best be understood as the outcome of the work of developing a coding scheme rather than as a stipulated pre-condition.

In qualitative analysis, the value of mutual exclusivity is often rejected outright as rigid systematicity. But Saldaña (2016) acknowledges that the use of many such "simultaneous codes" can be a sign of researcher indecisiveness. He also suggests that while many first cycle approaches to coding allow for multiple codes to be applied to the same piece of data, many second cycle approaches try to do work similar to that described for content analysis. That is, they take a hodgepodge of coding categories created during the first cycle of coding and try to organize and condense them onto a small set of non-overlapping categories that capture underlying relationships.

The process of code development that we outline in this book works toward mutual exclusivity as the foundation for a systematic and reliable analysis. In the rest of this chapter, we walk you through the process of building a coding scheme.

■ Getting Ready to Code

Developing a coding scheme involves an iterative process of moving back and forth between the developing scheme and a sample of the data to be coded.

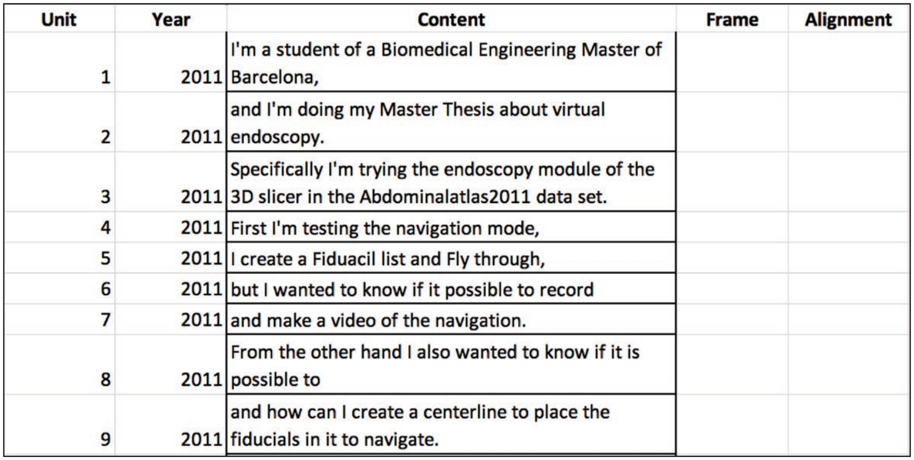
Selecting a Sample

Select parts of the stream of verbal data that can be easily coded in one sitting while at the same time giving you enough data to see all the relevant variations of your phenomenon. Your goal is to see enough of the data to develop a coding scheme that is comprehensive. Give some care to deciding how to select this initial sample. You will want to maximize the amount of variation in the phenomenon being investigated. In most situations, this means selecting a sizable amount of data from both sides of any contrasts you have built into your design.

From each side of the built-in contrast, try to select a part of the stream of verbal data that represents the best case for the phenomenon: From one side of the contrast, choose the part of the data in which the phenomenon is most obvious, or strongest. On the other side of the contrast, choose a part of the data in which you expect the phenomenon to be different. This initial comparison can be quite simple, for example choosing samples where the phenomenon might differ in terms of strength, sophistication, or focus. The aim of the comparison is to understand your built-in contrast a little better, especially in terms of how an emerging coding scheme might change along the lines of the contrast. Focusing on the differences between cases in your contrast will help you develop a coding scheme that highlights those differences.

How much data should you select? Select too little data and your coding scheme may not contain references to code categories that are relevant to the larger data set. Select too much data and you will code more than you need to establish the regularities you want to track. As a rule of thumb, you will want to select somewhere between 200 and 500 segments if they are shorter units (clauses, t-units, etc.) and significantly fewer if you are working with longer units (exchanges, topical units), maybe 50-100. Overall, you want to balance the amount to be read (10 pages seems about the most that someone can code initially), the number of segments to be coded, and the range of the phenomenon to be considered. Move each part of your contrasting sample into Excel or MAXQDA using the procedures described in Chapter 3.

In the Excel example to be used in this chapter, as shown in Figure 4.2, we begin with the data after it is segmented and moved to a spreadsheet. In this case, the data are taken from the email exchanges between users of a 3D imaging software package and are focused on the different ways that participants frame their discussion of the software. The data have been segmented using the clause as the unit of analysis. The data recorded for each segment of user action consist of the unit number (column A), the year posted to the list (column B), the writer (hidden in column C), and the clause itself (column D). Frame (column E) and Alignment (column f) represent the coding dimensions applied to this data set.



Unit	Year	Content	Frame	Alignment
1	2011	I'm a student of a Biomedical Engineering Master of Barcelona,		
2	2011	and I'm doing my Master Thesis about virtual endoscopy.		
3	2011	Specifically I'm trying the endoscopy module of the 3D slicer in the Abdominalatlas2011 data set.		
4	2011	First I'm testing the navigation mode,		
5	2011	I create a Fiduacil list and Fly through,		
6	2011	but I wanted to know if it possible to record		
7	2011	and make a video of the navigation.		
8	2011	From the other hand I also wanted to know if it is possible to		
9	2011	and how can I create a centerline to place the fiducials in it to navigate.		

Figure 4.2: Segmented data for coding.

Managing Coding

To manage the coding, always store your coding scheme in a way that is directly linked to the data that has been coded with that scheme. When you return to the data after days, months, or even years, this practice will alleviate the need to hunt for the scheme underlying an analysis. It will also enable you to keep track of which version of a coding scheme was used, with which set of coded data. Otherwise, proliferating coding schemes and multiple copies of data sets can make your life unbearable. See Excel Procedure 4.1 and MAXQDA Procedure 4.1.

Excel Procedure 4.1: Linking to Coding Scheme in Excel

https://goo.gl/5Q4Jgr

You can link coded data to its coding scheme in one of three ways:

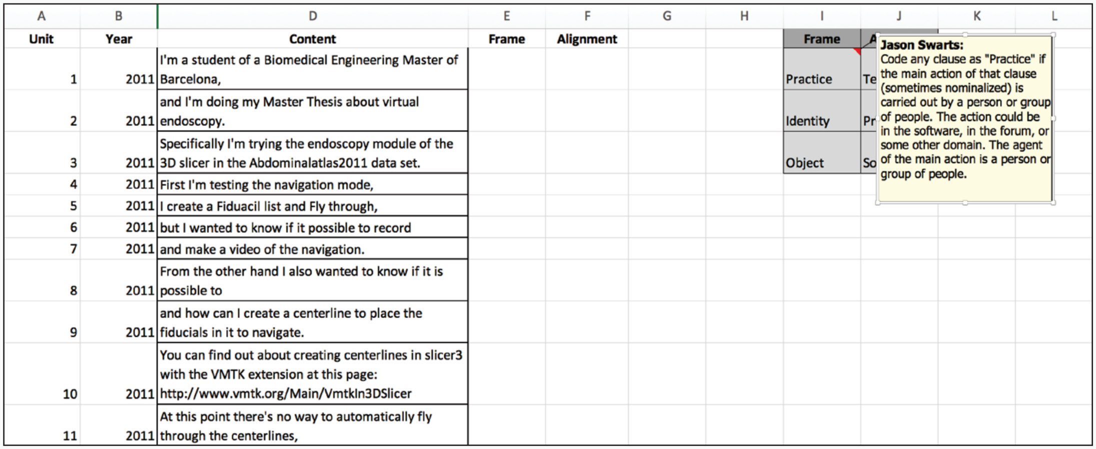
1. Write your coding scheme in a separate document file.
2. Link to it with a hyperlink from your Excel workbook.

Or

3. Write your coding scheme in a separate worksheet in your Excel workbook.

Or

4. Insert the coding categories and definitions as a comment using **Insert > New Comment** as shown in Figure 4.3.



A	B	D	E	F	G	H	I	J	K	L
Unit	Year	Content	Frame	Alignment			Frame	A		
1	2011	I'm a student of a Biomedical Engineering Master of Barcelona,					Practice	Te		
2	2011	and I'm doing my Master Thesis about virtual endoscopy.					Identity	Pr		
3	2011	Specifically I'm trying the endoscopy module of the 3D slicer in the Abdominalatlas2011 data set.					Object	So		
4	2011	First I'm testing the navigation mode,								
5	2011	I create a Fiduacil list and Fly through,								
6	2011	but I wanted to know if it possible to record								
7	2011	and make a video of the navigation.								
8	2011	From the other hand I also wanted to know if it is possible to								
9	2011	and how can I create a centerline to place the fiducials in it to navigate.								
10	2011	You can find out about creating centerlines in slicer3 with the VMTK extension at this page: http://www.vmtk.org/Main/VmtkIn3DSlicer								
11	2011	At this point there's no way to automatically fly through the centerlines,								

Jason Swarts:
Code any clause as "Practice" if the main action of that clause (sometimes nominalized) is carried out by a person or group of people. The action could be in the software, in the forum, or some other domain. The agent of the main action is a person or group of people.

Figure 4.3 Inserting coding definitions as comments in Excel.

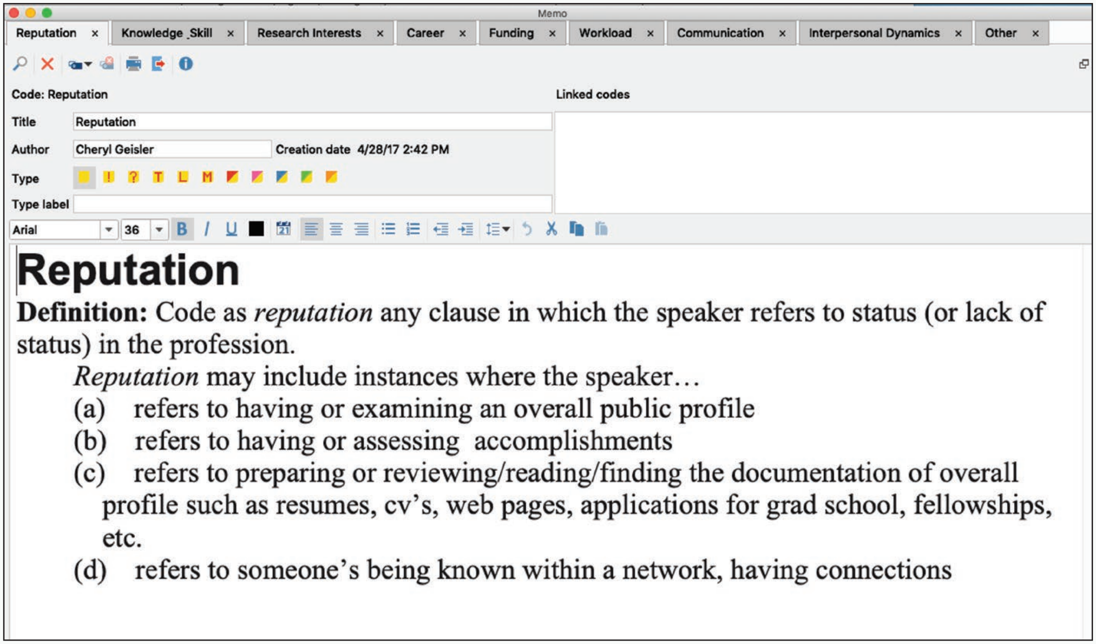
MAXQDA Procedure 4.1: Linking to Your Coding Scheme in MAXQDA

https://goo.gl/5Q4Jgr

In MAXQDA, your coding scheme is stored in the **Code System** window. Full definitions are stored as memos attached to each code name. To open the full coding scheme:

1. Double click on the **Memo** icon for each code in the order in which they are to be considered.

All of the relevant codes will apear in a tabbed **Memo** window as shown in Figure 4.4.



Code: Reputation

Title: Reputation

Author: Cheryl Geisler

Creation date 4/28/17 2:42 PM

Type label:

Reputation

Definition: Code as *reputation* any clause in which the speaker refers to status (or lack of status) in the profession.

Reputation may include instances where the speaker…

(a) refers to having or examining an overall public profile

(b) refers to having or assessing accomplishments

(c) refers to preparing or reviewing/reading/finding the documentation of overall profile such as resumes, cv's, web pages, applications for grad school, fellowships, etc.

(d) refers to someone's being known within a network, having connections

Figure 4.4 Opening the full coding scheme in MAXQDA.

Deciding on a Coding Framework

Two methods exist for generating a coding scheme. In the first, you begin with existing categories. Your review of the literature, while designing the analysis, might have suggested coding categories that characterize the phenomenon of interest. Or your work on the analytic design may have already suggested the ways in which the built-in contrasts in your data set should be different. And finally, especially if you have mucked around in a site, you may have strong intuitions about what to look for. All three of these sources—literature, built-in contrasts, and intuition—should be consulted to create a "start list" of possible categories that may be relevant to coding. You will, of course, need to extend and modify this start list as you go along, in order to mold your categories more appropriately to the data.

The second method for generating a coding scheme is to look at the initial data set and let it "speak to you." That is, you let each segment of the data suggest appropriate categories to describe what is happening with the phenomenon of interest. Such categories are more grounded in the data than those gleaned from external sources like the literature but cannot help but be influenced by the knowledge and experience you bring to the analysis.

Although in this second method of developing a coding scheme you are letting the data speak to you, the process of developing coding categories need not be without structure. Books on coding, discourse analysis, rhetorical theory, social theory, or others can provide useful interpretive frameworks for narrowing your investigation of the data based on what you suspect to be of interest in the phenomenon of study. For example, any of the following might be useful frameworks for investigating your data:

- Relationships Coding: the researcher pays attention to formal or informal connections between people or things, especially when the phenomenon of interest might be interpersonal or intrapersonal relationships (e.g., see Gee, 2005).
- Values Coding: the researchers pays attention to words and phrases that indicate values, attitudes, and beliefs expressed in the verbal data stream. This approach may be useful if the phenomenon of interest is thought to be influenced by belief systems or cultural values (Saldaña,

2016, pp. 131-132).

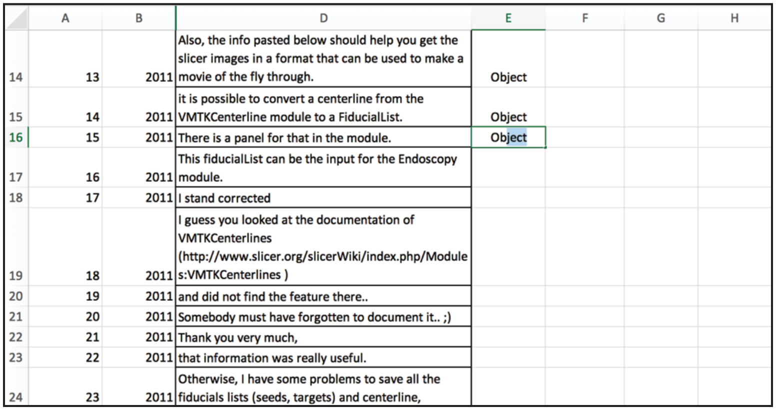
- Practices Coding: the researcher pays attention to words and phrases that indicate the presence of an emerging practice. This approach might be useful for seeing if a group of people interact as a community (e.g., see Wenger, 1998).
- Genre Coding: the researcher pays attention to words and phrases that define a conventional and habitual form of written or spoken communication. This approach might be useful for seeing attention to recurring work that is supported with written or spoken discourse (e.g., see Bazerman, 1988)

Choosing codes for exploring the data can be a matter of intuition or it can be suggested by the design of your analysis. Often you will try to code the data using more than one coding pass, until deciding on one that will lead to a productive analysis. The process can be time consuming, but it is better to spend that time at this stage of the research than to discover later on, in the analysis, that your coding scheme is not capturing characteristics of the phenomenon that are most interesting to you.

Finally, a somewhat less structured approach to letting the data speak to you would be using the results of a keyword or collocation analysis that we suggested in Chapter 2. The data resulting from a study of keywords in AntConc can sometimes point to clusters of words that have some shared "aboutness" (Scott, 1997) that could suggest a code. For example, studying interactions between tutors and students in a writing center might reveal keywords such as "tell" or "read" or "explain" which could suggest a focus on coding tutor actions.

■ Using a Coding Scheme

The process of developing a coding scheme, no matter how you started it, involves the same process (see Excel Procedure 4.2 and MAXQDA Procedure 4.2). Take each segment in your initial data sample and try to assign it one of the codes in your scheme. As long as such an assignment can be made, coding continues. You should assign each segment of data to one and only one category in the coding scheme. As shown in Figure 4.5, for example, each segment of the sample data has been coded for *Frame*. Segments 13, 14, and 15 all fit under the category of *Object* since they are describing or naming a feature of the software under discussion.



	A	B	D	E	F	G	H
14	13	2011	Also, the info pasted below should help you get the slicer images in a format that can be used to make a movie of the fly through.	Object			
15	14	2011	it is possible to convert a centerline from the VMTKCenterline module to a FiducialList.	Object			
16	15	2011	There is a panel for that in the module.	Object			
17	16	2011	This fiducialList can be the input for the Endoscopy module.				
18	17	2011	I stand corrected				
19	18	2011	I guess you looked at the documentation of VMTKCenterlines (http://www.slicer.org/slicerWiki/index.php/Modules:VMTKCenterlines)				
20	19	2011	and did not find the feature there..				
21	20	2011	Somebody must have forgotten to document it.. ;)				
22	21	2011	Thank you very much,				
23	22	2011	that information was really useful.				
24	23	2011	Otherwise, I have some problems to save all the fiducials lists (seeds, targets) and centerline,				

Figure 4.5 Assigning a code in Excel.

Excel Procedure 4.2: Assigning a Code in Excel

https://goo.gl/5Q4Jgr

To assign a code to a unit of data in Excel:

1. Label the coding column with the name of the coding dimension.
2. Position your cursor in the coding column in the cell that is adjacent to the unit to be coded.
3. Type in the name of the code you want to assign.

Once you have assigned a code, using it again is relatively easy with Excel's auto-completion.

4. Position your cursor in the coding column next to the unit to be coded.
5. Start typing the code.

As soon as you type the first few letters of a code you have used previously, Excel will automatically suggest that completion.

6. To select it, hit enter.

To use auto-completion effectively, it is best to use codes that are distinct in their first few letters.

Revising a Coding Scheme

When you encounter a segment that is not covered in the coding scheme, you need to revise your scheme. The process of revising your coding scheme should accomplish two things. First, the revision process helps you to come up with a set of coding categories that best reveal the distinctions that you consider important in tracking the phenomenon of interest. Second, and almost as important, the revision process should help you to understand the phenomenon of interest better—what is it that you are coding for? In the next few sections we talk about further development of coding schemes to reflect this growing understanding.

Clarifying a Definition

If you encounter a unit that you feel sure belongs in a specific coding category, but which does not clearly fit your working definition, you may need to clarify the coding definition. Coding definitions may be clarified by revising the cod-

MAXQDA Procedure 4.2: Assigning a Code in MAXQDA

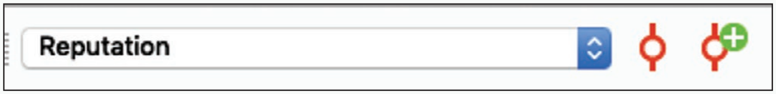
https://goo.gl/5Q4Jgr

Many different ways exist to assign codes in MAXQDA; here we cover just a few of them. You may want to consult the online documentation for more possibilities.

1. Select the paragraph number next to the unit to be coded.
2. Drag the selected unit onto the desired code in the list of codes found in the **Code System** window.

You will notice that the last code chosen is visible in the code drop down list in the **Document Browser** toolbar as shown in Figure 4.6. If you want to choose this code again,

3. Click on the **Code symbol** to the right of the drop down list in the **Document Browser** tool**bar**.



Reputation

Figure 4.6: Assigning a code in MAXQDA.

ing definition, by adding cases to indicate the kinds of phenomena to which a code applies, by adding examples to illustrate those cases—or all of the above. In general, clarification is the easiest way to revise a coding scheme and is the one you should consider before more complex revisions.

Adding a Category

If you feel that a unit does not fit into any of your existing categories, you may want to add a new category (see Excel Procedure 4.3 and MAXQDA Procedure 4.3). For example, when the coder reached Segment 36 in our example, she realized that the statement "Anything else would require programming" was not really a reference to a practice, but innovation of the software. So she added a new code *Innovation* to cover the case. Applying codes and generating new codes as needed continues until the sample of data taken for initial coding has been coded.

Breaking a Category Apart

Breaking a category apart is required when you realize that you have been lumping together phenomena that need to be distinguished. For example, suppose the coder has reached Segment 36, in which the user stated that "Anything else would require programming," and instead of deciding that this clause was a different kind of frame (*Innovation* as opposed to *Practice*) she decided that innovation was really just a different kind of *Practice* in that it is a way of interacting with the software. In this case, the *Practice* code would need to be broken apart.

Once you decide to break apart an existing code into components, begin by editing your coding scheme to remove the current category and replace it with codes for the new categories (see Excel Procedure 4.3 and MAXQDA Procedure 4.3). Next, review all data coded thus far and recode with the new codes using the procedure described below. Thus, if our coder decided to split the *Practice* code into different kinds of practices, she would need to add codes for *Innovation* and other kinds of practice that are differentiated from *Innovation* (e.g., *Update*, *Use*, etc.). She would then revise the coding for all of the segments coded so far, accounting for the new coding scheme.

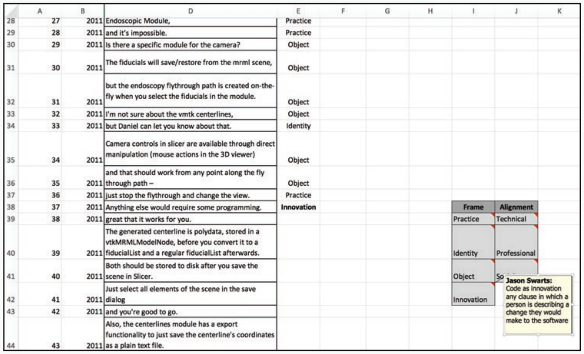
Excel Procedure 4.3: Creating a Code in Excel

https://goo.gl/5Q4Jgr

To create a new code in Excel, as shown in Figure 4.7:

1. Add a new code to your coding scheme.
2. After the code name include the unit of analysis to which it applies, and a definition of the coding category in the following form:

   Code as [code name] any [unit] that [definition].



	A	B	D	E
28	27	2011	Endoscopic Module,	Practice
29	28	2011	and it's impossible.	Practice
30	29	2011	Is there a specific module for the camera?	Object
31	30	2011	The fiducials will save/restore from the mrml scene,	Object
32	31	2011	but the endoscopy flythrough path is created on-the-fly when you select the fiducials in the module.	Object
33	32	2011	I'm not sure about the vmtk centerlines,	Object
34	33	2011	but Daniel can let you know about that.	Identity
35	34	2011	Camera controls in slicer are available through direct manipulation (mouse actions in the 3D viewer)	Object
36	35	2011	and that should work from any point along the fly through path –	Object
37	36	2011	just stop the flythrough and change the view.	Practice
38	37	2011	Anything else would require some programming.	**Innovation**
39	38	2011	great that it works for you.	
40	39	2011	The generated centerline is polydata, stored in a vtkMRMLModelNode, before you convert it to a fiducialList and a regular fiducialList afterwards.	
41	40	2011	Both should be stored to disk after you save the scene in Slicer.	
42	41	2011	Just select all elements of the scene in the save dialog	
43	42	2011	and you're good to go.	
44	43	2011	Also, the centerlines module has a export functionality to just save the centerline's coordinates as a plain text file.	

Frame	Alignment
Practice	Technical
Identity	Professional
Object	So
Innovation	

Jason Swarts:
Code as innovation any clause in which a person is describing a change they would make to the software

Figure 4.7 Creating a code within Excel.

You need not add cases and examples at this point, but you may want to as you encounter them.

MAXQDA Procedure 4.3: Creating a Code in MAXQDA

https://goo.gl/5Q4Jgr

To create a new code in MAXQDA:

1. Click on the code symbol with the plus sign in the **Document Browser** toolbar.
2. Type the name of the code into the resulting dialog window.
3. Use the **Memo** field to write a definition in the following form:

   Code as [code name] any [unit] that [definition].

Your new code will now be listed in the **Code System** window.

Be careful about how you split categories into sub-components. Make sure that the new distinctions that you are entering into your coding scheme are both relevant to your phenomenon of interest and at the same level as the remaining codes in your scheme. In other words, codes within a single dimension should have a parallelism to them. For example, once our coder splits the *Practice* code, she may begin to wonder whether the remaining codes in her coding scheme, also need further refinement. If there are multiple kinds of *Practice*, perhaps the same is true of *Object* and *Identity*. Deciding whether to take this course of action, however, requires you to consider the net benefit to analysis gained by making your coding scheme more complicated.

Collapsing Categories

At other times, you may find that working up to a coding scheme from early exploratory coding leaves you with too many coding categories, not all of which are distinct enough to stand alone. If you find that there are some segments of data that are not easily or definitively placed in a particular coding category, it may be worthwhile to collapse coding categories together.

Before taking this step, verify that two other kinds of corrections might not work better. First, consider whether there is some ambiguity in the wording of two code definitions that could be clarified in a way that makes coding of all segments clearer. Second, consider how often you encounter the segments that are unclear. If appropriate, you can make a decision about segments that are unclear and specify that they should be included in one coding category over the other. If neither of these fixes solves the problem, you may need to collapse codes together.

The advantage of combining codes is that the coding decisions become a little bit easier since the code category is larger and more inclusive. Less coder discrimination among segments is now required. The clear disadvantage is that in making coding decisions easier, the coder loses some ability to tease apart one code from another. This is only a problem if that distinction matters to the analysis. In some cases, it may not. If you do combine codes, the same reconstructive work required for breaking code categories apart also applies to collapsing categories. The coder must go back through the previously coded work and collapse together codes that were once separate.

Re-coding Data

When you decide to revise, add, or collapse codes in your coding scheme, you will need to recode the data you have coded thus far (see Excel Procedure 4.4, MAXQDA Procedure 4.4, and MAXQDA Procedure 4.5). Going through the data a second time with new coding categories in mind can help you to spot the consequences of your revisions. It is often the case that a new or revised code affects far more data than you originally expect.

Excel Procedure 4.4: Re-coding Data in Excel

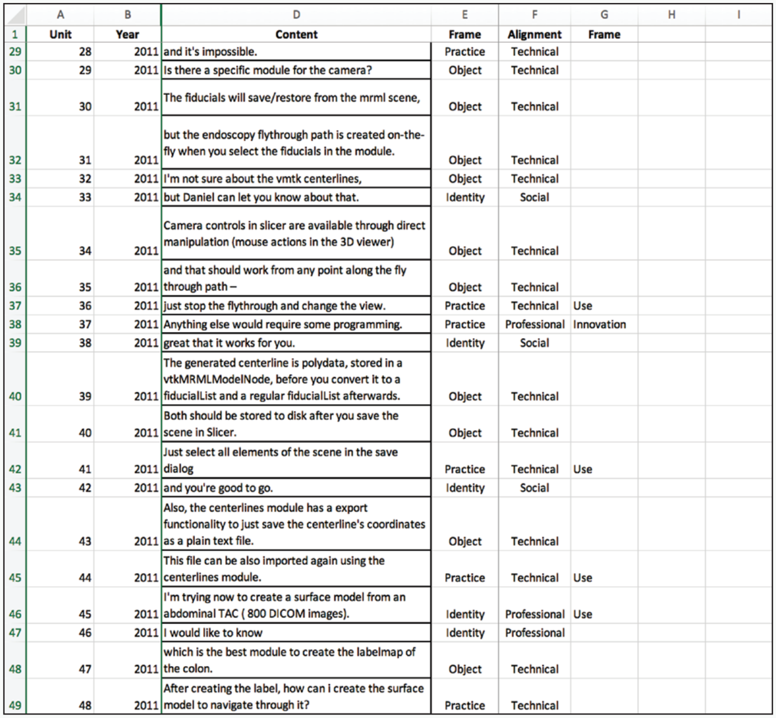
https://goo.gl/5Q4Jgr

To try out a new coding scheme:

1. Add a new coding column as shown in Figure 4.8.
2. Use it to apply your new coding categories.

Which coding scheme better gets at what you are interested in?

It is not at all uncommon for a coder to decide that the first scheme is better than the more detailed scheme. If this is the case, then keeping rather than deleting the first coding will save a great deal of reconstructive work.



	A	B	D	E	F	G	H	I
1	Unit	Year	Content	Frame	Alignment	Frame		
29	28	2011	and it's impossible.	Practice	Technical			
30	29	2011	Is there a specific module for the camera?	Object	Technical			
31	30	2011	The fiducials will save/restore from the mrml scene,	Object	Technical			
32	31	2011	but the endoscopy flythrough path is created on-the-fly when you select the fiducials in the module.	Object	Technical			
33	32	2011	I'm not sure about the vmtk centerlines,	Object	Technical			
34	33	2011	but Daniel can let you know about that.	Identity	Social			
35	34	2011	Camera controls in slicer are available through direct manipulation (mouse actions in the 3D viewer)	Object	Technical			
36	35	2011	and that should work from any point along the fly through path –	Object	Technical			
37	36	2011	just stop the flythrough and change the view.	Practice	Technical	Use		
38	37	2011	Anything else would require some programming.	Practice	Professional	Innovation		
39	38	2011	great that it works for you.	Identity	Social			
40	39	2011	The generated centerline is polydata, stored in a vtkMRMLModelNode, before you convert it to a fiducialList and a regular fiducialList afterwards.	Object	Technical			
41	40	2011	Both should be stored to disk after you save the scene in Slicer.	Object	Technical			
42	41	2011	Just select all elements of the scene in the save dialog	Practice	Technical	Use		
43	42	2011	and you're good to go.	Identity	Social			
44	43	2011	Also, the centerlines module has a export functionality to just save the centerline's coordinates as a plain text file.	Object	Technical			
45	44	2011	This file can be also imported again using the centerlines module.	Practice	Technical	Use		
46	45	2011	I'm trying now to create a surface model from an abdominal TAC (800 DICOM images).	Identity	Professional	Use		
47	46	2011	I would like to know	Identity	Professional			
48	47	2011	which is the best module to create the labelmap of the colon.	Object	Technical			
49	48	2011	After creating the label, how can i create the surface model to navigate through it?	Practice	Technical			

Figure 4.8: Adding a trial coding column in Excel.

MAXQDA Procedure 4.4: Re-coding Data in MAXQDA

https://goo.gl/5Q4Jgr

In MAXQDA, changes in your coding scheme will require you to review and possibly recode the data.

1. Double click the **Memo** icon next to the code you want to revise.
2. Edit the definition to make changes.
3. Read through your already coded data using this newly revised definition.
4. To delete a code from a segment, right click on the code and select delete.
5. To add a code to a segment, click on the segment number and apply the new code.

MAXQDA Procedure 4.5: Collapsing Coding Categories in MAXQDA

https://goo.gl/5Q4Jgr

If you want to collapse two categories into a new code in your coding scheme, make them into subcodes of the new code:

1. Click on the **New Code** icon in the **Code System** window.
2. Name and define the new code.
3. Drag the desired subcodes on top of the new code in the **Code System** window.

As shown in Figure 4.9, the dragged code becomes a subcode of the main code.

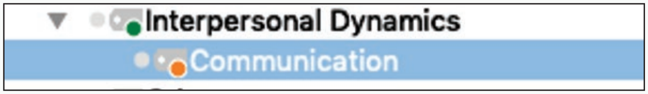


Figure 4.9: Making a subcode in MAXQDA.

Adding Another Dimension

A related problem to collapsing coding categories is one where a coder finds the need to apply two different codes to the same segment. As we have already said, a scheme should be so devised that the codes are mutually exclusive. If the two codes that you want to apply are of the same type and do belong to the same dimension, consider collapsing the codes together (e.g., *Use* and *Innovation*) but if the codes are different kinds of observations, then it may indicate that you are dealing with two dimensions of the phenomenon that ought to be coded separately.

A dimension is a range of variation, presented through a coding scheme, representing an aspect of the data that can stand as conceptually independent of other features. Each coding scheme should be associated with only one dimension of data variation that, in turn, should correspond to only one feature of the phenomenon of interest. When you realize that you are dealing with two distinct dimensions of your data, you should separate them, develop a separate coding scheme for each one, and then apply each scheme to the data separately (see Excel Procedure 4.5 and MAXQDA Procedure 4.6).

Excel Procedure 4.5: Adding a Dimension in Excel

https://goo.gl/5Q4Jgr

To add a dimension in Excel:

1. Create a new second coding scheme.
2. Create a new coding column in your data worksheet.

The new coding scheme should have all of the elements of a good scheme: the name of the new dimension, its coding categories, the unit of analysis, and at least a definition of each category.

You will most likely be moving a coding definition into your new scheme from your original coding scheme. But you will probably add other categories as well.

Each of your coding schemes should be developed and applied to the data in a separate pass. It is too much to ask a coder to work with more than one dimension at a time.

For example, when our coder wondered why *Innovation* should have its own code while *Use* does not, she could have begun to think about separating her analysis into two distinct dimensions. She might have realized that the code *Practice* did not so much represent a different kind of frame as much as a distinct way of engaging the software. For this reason, rather than revising the coding scheme for frame, then, our coder could instead create a new dimension for coding called Engagement and develop a coding scheme to track it. In this example, our coder would code the entire sample of data for Frame first and then return to do a second pass through the data on this new dimension of Engagement.

MAXQDA Procedure 4.6: Adding a Dimension in MAXQDA

https://goo.gl/5Q4Jgr

The best way to add a dimension in MAXQDA is to reorganize your codes into separate Code Sets, each of which represents a separate dimension. To create a new Code Set, as shown in Figure 4.10:

1. Right click on **Sets** in the **Code System** window.

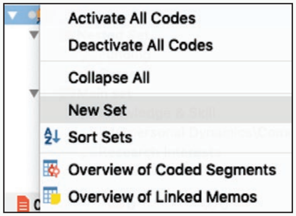


Figure 4.10: Creating a new code set in MAXQDA.

2. Choose the option to create a **New Set** and name it with the name of your first dimension.
3. Then drag and drop into the new set the appropriate codes from your **Code System** list.
4. To create your new second dimension, create a second **Code Set** in the same way.

To focus on a single dimension for coding:

5. Right click in the grey area of the coding column in the **Document Browser**. Choose the option **Only activated codes**.
6. Return to the **Code System** window and activate the current dimension's codes by clicking the small grey circles in front of the dimension's name.

Exercise 4.1 Test Your Understanding

Open any piece of writing from any source and segment that content into sentences. Use the coding scheme below to code that data. Pay attention to problems that arise in using this coding scheme, note those problems and discuss how to fix them.

- **Information:** Code as Information any sentence in which the author is reporting factual content.
- **Description:** Code as Description any sentence in which the author is relying on sensory information to create a mental picture for the reader.
- **Persuasion:** Code as Persuasion any sentence in which the author is using reasoning and evidence to advance a point.
- **Explanation:** Code as Explanation any sentence in which the author is making something clear by adding detail and motivation.
- **Humor:** Code as Humor any sentence in which the author is attempting to be funny.
- **Narration:** Code as Narration any sentence in which the author is telling a story.

For Discussion: What are the problems that you discovered when using this coding scheme? What are the different approaches for dealing with those problems?

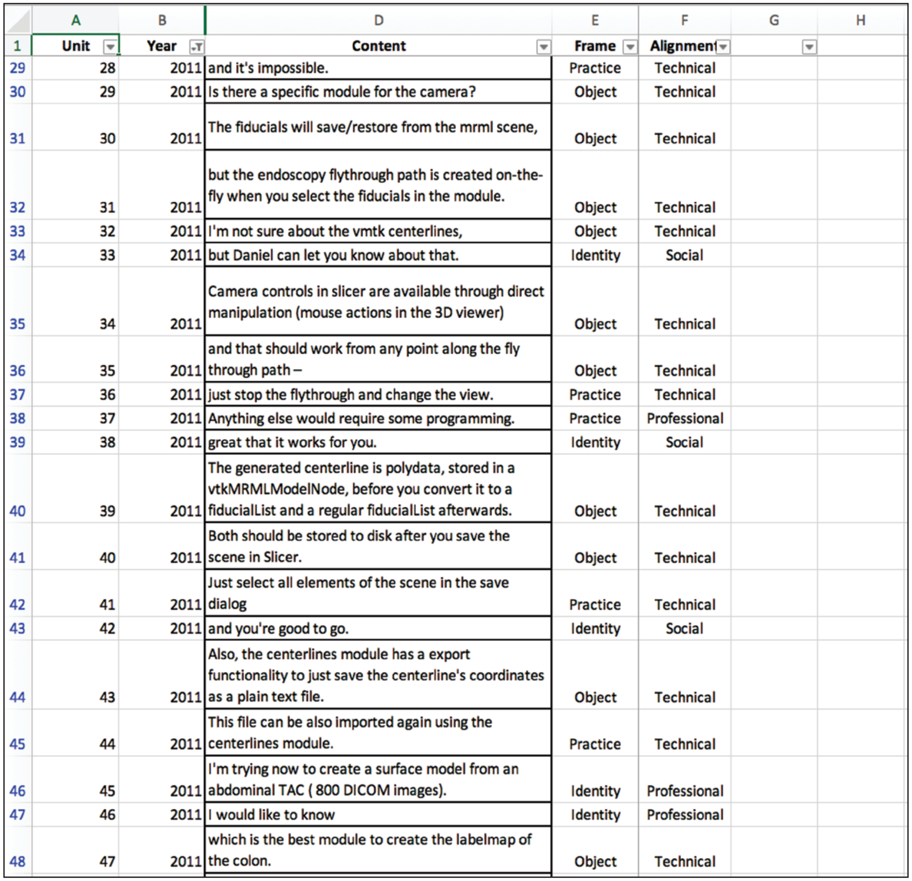
Memo 4.1 Dimensions of Analysis

Reflect on your phenomenon of study and the qualities about it that interest you. Make a list of those qualities and begin sorting them into categories of like items. What are the potential dimensions of your analysis and what are the categories that belong to those dimensions?

Techniques for Inspecting Coding

As you develop a coding scheme during the iterative back and forth between data and scheme, you will often want to look at all of the data to which you have assigned a specific code. You can then check to see whether your coding decisions have been consistent (see Excel Procedure 4.6 and MAXQDA Procedure 4.7).

The most important technique for inspecting codes in Excel is Filtering. When you ask Excel to filter your data, you ask it to show you data that meets certain criteria and hide the rest. In Figure 4.11, for example, a Filter has been activated in Column B showing all data that has been coded as referring to units posted in 2011.



	A	B	D	E	F	G	H
1	Unit	Year	Content	Frame	Alignment		
29	28	2011	and it's impossible.	Practice	Technical		
30	29	2011	Is there a specific module for the camera?	Object	Technical		
31	30	2011	The fiducials will save/restore from the mrml scene,	Object	Technical		
32	31	2011	but the endoscopy flythrough path is created on-the-fly when you select the fiducials in the module.	Object	Technical		
33	32	2011	I'm not sure about the vmtk centerlines,	Object	Technical		
34	33	2011	but Daniel can let you know about that.	Identity	Social		
35	34	2011	Camera controls in slicer are available through direct manipulation (mouse actions in the 3D viewer)	Object	Technical		
36	35	2011	and that should work from any point along the fly through path –	Object	Technical		
37	36	2011	just stop the flythrough and change the view.	Practice	Technical		
38	37	2011	Anything else would require some programming.	Practice	Professional		
39	38	2011	great that it works for you.	Identity	Social		
40	39	2011	The generated centerline is polydata, stored in a vtkMRMLModelNode, before you convert it to a fiducialList and a regular fiducialList afterwards.	Object	Technical		
41	40	2011	Both should be stored to disk after you save the scene in Slicer.	Object	Technical		
42	41	2011	Just select all elements of the scene in the save dialog	Practice	Technical		
43	42	2011	and you're good to go.	Identity	Social		
44	43	2011	Also, the centerlines module has a export functionality to just save the centerline's coordinates as a plain text file.	Object	Technical		
45	44	2011	This file can be also imported again using the centerlines module.	Practice	Technical		
46	45	2011	I'm trying now to create a surface model from an abdominal TAC (800 DICOM images).	Identity	Professional		
47	46	2011	I would like to know	Identity	Professional		
48	47	2011	which is the best module to create the labelmap of the colon.	Object	Technical		

Figure 4.11: Using a filter to inspect coding in Excel.

Excel Procedure 4.6: Inspecting by Code in Excel

https://goo.gl/5Q4Jgr

To inspect by code in Excel:

1. Select the column containing the coding (Column B in this case).
2. Click **Data > Filter** from the toolbar.

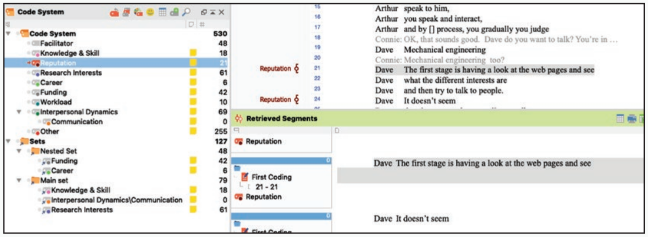
Dropdown arrows will appear at the top of the column (Figure 4.11).

3. Click on the dropdown arrows and unselect the option **Select all**.
4. Then select the code that you would like to inspect (in this case 2011).
5. To turn the filter off, return to the **Filter** command under **Data** in the toolbar and deselect it.

MAXQDA Procedure 4.7: Inspecting by Code in MAXQDA

https://goo.gl/5Q4Jgr

Inspecting data by code is accomplished in MAXQDA in the **Retrieved Segments** window as shown in Figure 4.12.



Code System	#
Code System	530
Facilitator	48
Knowledge & Skill	18
Reputation	21
Research Interests	61
Career	6
Funding	42
Workload	10
Interpersonal Dynamics	69
Communication	0
Other	255
Sets	127
Nested Set	48
Funding	42
Career	6
Main set	79
Knowledge & Skill	18
Interpersonal Dynamics\Communication	0
Research Interests	61

```
15            Arthur   speak to him,
16            Arthur   you speak and interact,
17            Arthur   and by [] process, you gradually you judge
18            Connie: OK, that sounds good.  Dave do you want to talk? You're in …
19            Dave     Mechanical engineering
20            Connie: Mechanical engineering  too?
Reputation 21 Dave     The first stage is having a look at the web pages and see
22            Dave     what the different interests are
23            Dave     and then try to talk to people.
Reputation 24 Dave     It doesn't seem
```

Reputation

First Coding
21 - 21
Reputation

Dave The first stage is having a look at the web pages and see

Dave It doesn't seem

First Coding

Figure 4.12: Retrieving data by the code Reputation in the Retrieved Segments window.

To retrieve data by code in MAXQDA:

1. Activate the document or documents you want to inspect in the **Document System** window.
2. Activate the code you want to inspect in the **Code System** window.
3. Scroll through the retrieved segments in the **Retrieved Segments** window.
4. To examine the segment in full context, click on the source information to its left in the **Retrieved Segments** window and the full context will appear in the **Document Browser** window.

Techniques for Automated Coding

Up to this point, we have largely been talking about using manual coding techniques for exploring your data. Working from a coding scheme, you apply a code category to each segment of data and then investigate the patterns. What you have likely noticed, both in developing and applying your code scheme, is that sometimes key terms or phrases seem to accompany your codes. If you are coding for hypothetical statements, for example, you might notice the presence of terms like "may" and "might" or phrases like "perhaps we can" or "maybe if" and recognizing these patterns can be the key to unlocking the potential of automated coding techniques to supplement the manual coding we have been discussing in this chapter so far. What follows is a brief overview of different ways to inspect your data for keywords in AntConc that might yield insights that lend themselves to automated coding.

Identifying Keywords in AntConc

The way that AntConc generates a list of keywords is by analyzing words and their frequencies in a study corpus by comparing it to a reference corpus. Often, a researcher will study keywords by choosing a reference corpus that offered some useful contrast with the study corpus. For example, one studying corporate apology letters might collect a sample of apology letters as a study corpus and then download an existing corpus of common business English to use as a reference corpus. Comparing the study corpus to the reference corpus would reveal words used in the apologies that are different from "normal" business English because of their unusual frequency. In our example, a comparison of corporate apologies against common business English might reveal the unusual frequency of words like "ensure" and "promise" and the way to interpret this result is that terms like "ensure" and "promise" are unique to the corporate apologies. They are key to understanding what apologies are about and how they differ from the "normal" business English to which they are compared. A simple keyword analysis can reveal a number of words that have keyness value, but not all matter for your analysis.

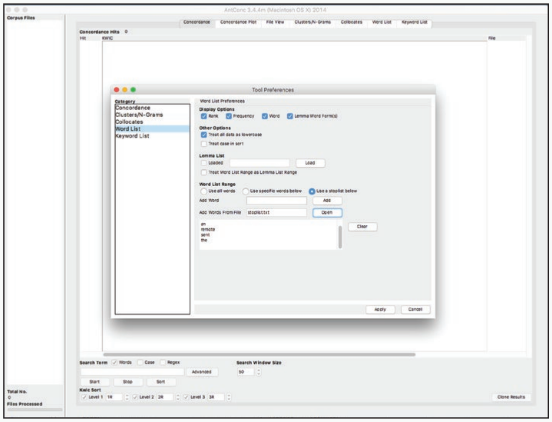
Although there are some free corpora for download (e.g., Supreme Court Decisions, Wikipedia) one need not go to such lengths to find a suitable reference corpus—you already have one in your built-in contrast. The data from your built-in contrast ought to be a sample of discourse that is akin to the discourse that you want to study but is different in that the phenomenon you want to study appears differently. This quality makes the contrast data a suitable reference corpus because it will highlight in the study corpus words that may be directly associated with the phenomenon that you want to study.

Load in your reference corpus and study corpus to AntConc following the instructions laid out in Chapter 2. To find keywords in the study corpus, begin by uploading a stopword list (see Procedure 4.1), which you can generate from a list of common stop words that is easily found on the internet. Just be careful because some stop lists will include common function words that may be important to your analysis (e.g., conditionals, modals, indexicals).

Procedure 4.1: Adding a Stop List in AntConc

https://goo.gl/5Q4Jgr

1. Click **Settings** > **Tool Preferences** and then click the **Word List** tool (Figure 4.13).
2. Add words in the **Add Word** field or upload a .txt file with a list of stop words using the **Add Words From File** field.



Tool Preferences

Category: Word List

Word List Preferences

Display Options: ☑ Rank ☑ Frequency ☑ Word ☑ Lemma Word Form(s)

Other Options: ☑ Treat all data as lowercase □ Treat case in sort

Lemma List: □ Loaded □ Treat Word List Range as Lemma List Range

Word List Range: ○ Use all words ○ Use specific words below ● Use a stoplist below

Add Words From File: stoplist.txt

an
remote
sent
the

Figure 4.13 Adding a stop word list in AntConc.

Now you are ready to review your study corpus for a list of keywords.

Keyness is a measure of how unusually frequent the term is in your study corpus. Generally, a word with a keyness rating (Log-Likelihood, which is the default measure) higher than 3.84 is considered significant enough for further inspection (https://www.lancaster.ac.uk/fss/courses/ling/corpus/blue/l08_4.htm).

With this list of keywords to examine, your job is to investigate words that have some affinity to them and that match to a coding category in your coding scheme. For example, if we had a coding scheme for analyzing corporate apologies that used a dimension for classifying types of statements, a keyword list generated from a corpus of apology letters might reveal words like "ensure," "promise," and "unwavering" which seem to point to statements of commitment. It might also be the case that there are more collective references like "we," "us," and "team" which seem to indicate statements of shared responsibility. If these words have high keyness ratings then this inspection will show that those statements are important to understanding the "aboutness" of those apologies. In effect, the keyness analysis is a check on the meaningfulness of the codes you have incorporated into your scheme (see Procedure 4.2).

■ Examining Keyword Clusters in AntConc

We can also inspect the meaningfulness of our codes by looking at keywords in context. Sometimes seeing the words that the keywords are associated with can give a richer understanding of the "aboutness" of those terms, how they are used in the stream of language that you are studying. This further step can help you decide if those keywords are pointing to the phenomenon that you are most interested in studying.

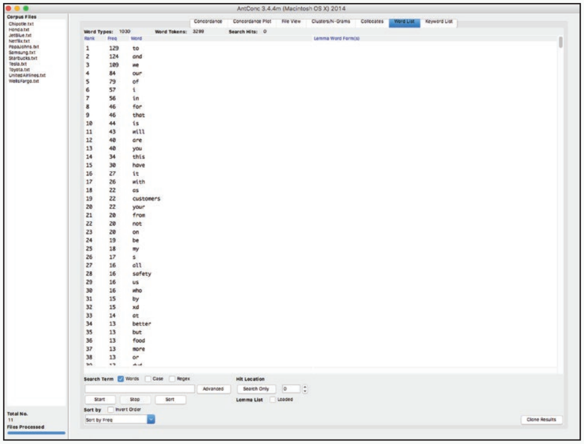
The clusters that are returned in a search will give you a better indication of ways that your keywords are used in context and will help you verify if the keywords in a text are both aligning with your code categories and with the phenomenon of interest in the data (see Procedures 4.3, 4.4, and 4.5).

In the end, your inspection of keywords one at a time in the Concordance and Cluster tools will reveal to you which terms coincide best with the coding categories you have developed and will show you which terms and code categories offer you the most traction in analyzing your data set. Create lists of associated key terms for use in automated coding, which we cover in the next section.

Procedure 4.2: Generating Keywords in AntConc

https://goo.gl/5Q4Jgr

1. From the main AntConc screen, click the **Word List** tab and then **Start** to generate a list of words in your study corpus.
2. The list generated will show all words, minus stop words, grouped and ordered by frequency (Figure 4.14).
3. Click on the **Keyword List** tab and then **Start** for a list of keywords sorted by keyness. (Figure 4.15)



Word Types: 1030 Word Tokens: 3299 Search Hits: 0

Rank	Freq	Word
1	129	to
2	124	and
3	109	we
4	84	our
5	79	of
6	57	i
7	56	in
8	46	for
9	46	that
10	44	is
11	43	will
12	40	are
13	40	you
14	34	this
15	30	have
16	27	it
17	26	with
18	22	as
19	22	customers
20	22	your
21	20	from
22	20	not
23	20	on
24	19	be
25	18	my
26	17	s
27	16	all
28	16	safety
29	16	us
30	16	who
31	15	by
32	15	xd
33	14	at
34	13	better
35	13	but
36	13	food
37	13	more
38	13	or

☑ Words □ Case □ Regex

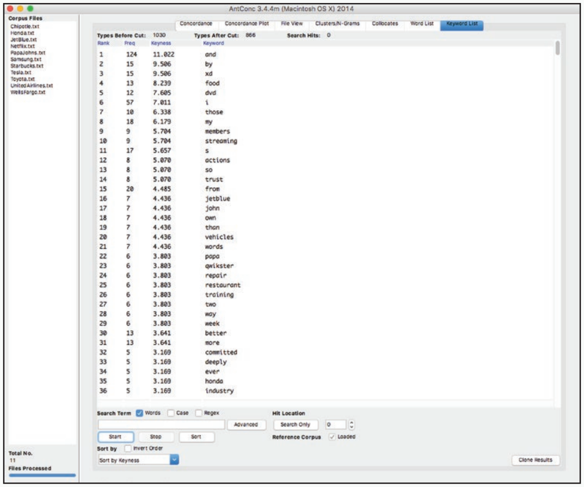
Hit Location: 0

□ Loaded

Sort by □ Invert Order: Sort by Freq

Total No. 11

Figure 4.14: Word list function in AntConc.



Types Before Cut: 1030 Types After Cut: 866 Search Hits: 0

Rank	Freq	Keyness	Keyword
1	124	11.022	and
2	15	9.506	by
3	15	9.506	xd
4	13	8.239	food
5	12	7.605	dvd
6	57	7.011	i
7	10	6.338	those
8	18	6.179	my
9	9	5.704	members
10	9	5.704	streaming
11	17	5.657	s
12	8	5.070	actions
13	8	5.070	so
14	8	5.070	trust
15	20	4.485	from
16	7	4.436	jetblue
17	7	4.436	john
18	7	4.436	own
19	7	4.436	than
20	7	4.436	vehicles
21	7	4.436	words
22	6	3.803	papa
23	6	3.803	qwikster
24	6	3.803	repair
25	6	3.803	restaurant
26	6	3.803	training
27	6	3.803	two
28	6	3.803	way
29	6	3.803	week
30	13	3.641	better
31	13	3.641	more
32	5	3.169	committed
33	5	3.169	deeply
34	5	3.169	ever
35	5	3.169	honda
36	5	3.169	industry

☑ Words □ Case □ Regex

Hit Location: 0

✓ Loaded

Sort by □ Invert Order: Sort by Keyness

Total No. 11

Figure 4.15 Keywords found in the study corpus.

Procedure 4.3: Examining Keywords in Context

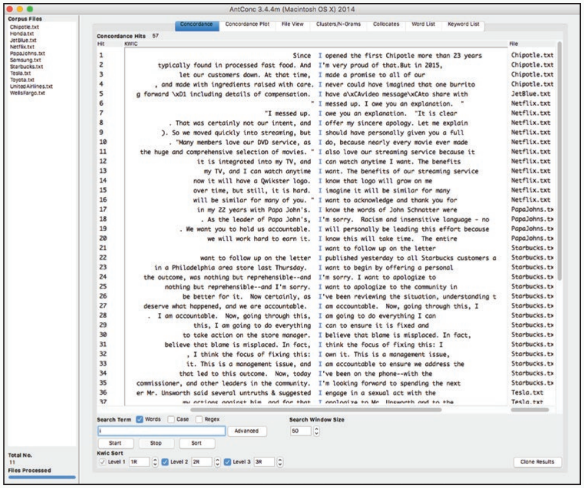
https://goo.gl/5Q4Jgr

1. From the keyword list on the **Keyword** tab, click on any word that is of interest.

AntConc will redirect you to the **Concordance** tab, which will show a list of references to those key terms in context (Figure 4.16). You will see the word, the sentence that it appears in, and the file in which that sentence occurs.

2. Inspect the keyword in context.

Inspecting in this way can help you see how prevalent each keyword might be across the cases in the study corpus. The view will also let you see how frequently the term occurs and the contextual information will allow you to determine if the word is both a good indicator of a particular code category and if the code category points to something meaningful about Keyness in our verbal data phenomenon.



Chipotle.txt
Honda.txt
JetBlue.txt
Netflix.txt
PapaJohns.txt
Samsung.txt
Starbucks.txt
Tesla.txt
Toyota.txt
UnitedAirlines.txt
WellsFargo.txt

Total No. 11

Concordance Hits 57

```
Hit  KWIC                                                                                                                        File
1                                                  Since  I opened the first Chipotle more than 23 years              Chipotle.txt
2        typically found in processed fast food. And  I'm very proud of that.But in 2015,                         Chipotle.txt
3                   let our customers down. At that time,  I made a promise to all of our                          Chipotle.txt
4        , and made with ingredients raised with care. I never could have imagined that one burrito                Chipotle.txt
5       g forward \xD1 including details of compensation.  I have a\xCAvideo message\xCAto share with             JetBlue.txt
6                                                      " I messed up. I owe you an explanation.  "                 Netflix.txt
7                                           "I messed up.  I owe you an explanation.  "It is clear                Netflix.txt
8               . That was certainly not our intent, and  I offer my sincere apology. Let me explain               Netflix.txt
9             ). So we moved quickly into streaming, but  I should have personally given you a full               Netflix.txt
10             . "Many members love our DVD service, as  I do, because nearly every movie ever made              Netflix.txt
11   the huge and comprehensive selection of movies. " I also love our streaming service because it           Netflix.txt
12                   it is integrated into my TV, and  I can watch anytime I want. The benefits                Netflix.txt
13                    my TV, and I can watch anytime  I want. The benefits of our streaming service          Netflix.txt
14                  now it will have a Qwikster logo.  I know that logo will grow on me                       Netflix.txt
15                  over time, but still, it is hard.  I imagine it will be similar for many                 Netflix.txt
16                will be similar for many of you. " I want to acknowledge and thank you for              Netflix.txt
17                   in my 22 years with Papa John's.  I know the words of John Schnatter were             PapaJohns.tx
18                     . As the leader of Papa John's,  I'm sorry.  Racism and insensitive language - no   PapaJohns.tx
19              . We want you to hold us accountable.  I will personally be leading this effort because   PapaJohns.tx
20                      we will work hard to earn it.  I know this will take time.  The entire            PapaJohns.tx
21                                                      I want to follow up on the letter                 Starbucks.tx
22                  want to follow up on the letter  I published yesterday to all Starbucks customers a  Starbucks.tx
23        in a Philadelphia area store last Thursday.   I want to begin by offering a personal           Starbucks.tx
24   the outcome, was nothing but reprehensible--and  I'm sorry. I want to apologize to                 Starbucks.tx
25         nothing but reprehensible--and I'm sorry.  I want to apologize to the community in           Starbucks.tx
26               be better for it.  Now certainly, as  I've been reviewing the situation, understanding t Starbucks.tx
27      deserve what happened, and we are accountable.   I am accountable.  Now, going through this, I  Starbucks.tx
28     .  I am accountable.  Now, going through this,  I am going to do everything I can               Starbucks.tx
29                this, I am going to do everything  I can to ensure it is fixed and                    Starbucks.tx
30               to take action on the store manager.  I believe that blame is misplaced. In fact,      Starbucks.tx
31          believe that blame is misplaced. In fact,  I think the focus of fixing this: I             Starbucks.tx
32                  , I think the focus of fixing this:  I own it. This is a management issue,          Starbucks.tx
33               it. This is a management issue, and  I am accountable to ensure we address the       Starbucks.tx
34                that led to this outcome.  Now, today  I've been on the phone--with the             Starbucks.tx
35  commissioner, and other leaders in the community.  I'm looking forward to spending the next       Starbucks.tx
36  er Mr. Unsworth said several untruths & suggested  I engage in a sexual act with the              Tesla.txt
37              my actions against him, and for that  I apologize to Mr. Unsworth and to the        Tesla.txt
```

Search Term ☑ Words ☐ Case ☐ Regex

i [Advanced]

Search Window Size 50

Kwic Sort ☑ Level 1 1R ☑ Level 2 2R ☑ Level 3 3R

Figure 4.16 Seeing keywords in context in AntConc.

Procedure 4.4: Examining a List of Keywords in Context

https://goo.gl/5Q4Jgr

If you want to see an entire list of keywords in context, it is also possible to do that.

1. On the **Keyword** tab, type in one of your keywords in the search field and then click **Advanced**.
2. On the **Advanced** screen check the option for **Use search term(s)** from list below and enter the remaining keywords, each on its own line.
3. Click **Apply** and return to the **Cluster** tool.
4. Click **Start**.

The list returned will show the phrases for all of your keywords.

Procedure 4.5: Examining Keyword Custers

https://goo.gl/5Q4Jgr

You can further refine the code categories by looking at the word clusters that keywords appear in (Figure 4.17).

1. Click on the **Cluster/N-gram** tool.
2. Check the option **Search Term: Word**.

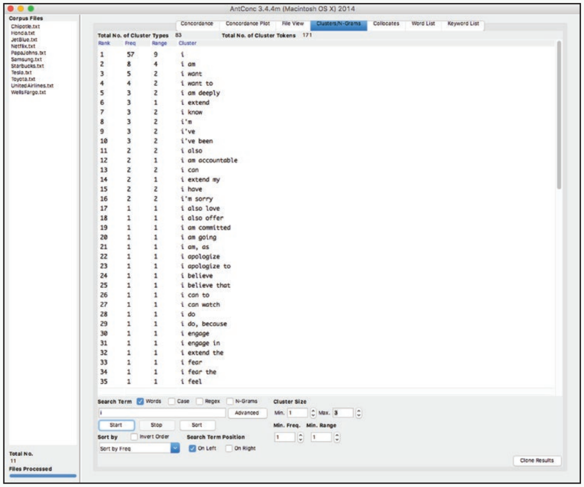
The **Cluster** tool will return phrases in which your searched keyword appears.

3. Set the parameters of the search by changing the **cluster size**, to designate how many words to the left and right to include in the cluster around your key term.
4. Set the minimum frequency (**Min.Freq.**) and minimum range (**Min. Range**) for your results.

Minimum frequency sets a lower threshold for how many times a cluster needs to appear in the study corpus before it is returned as a hit. Set this number fairly low to start.

The **minimum range** sets the lower threshold for the number of files or samples that the cluster must appear in to be returned in the results. You should also set this number fairly low to start.

5. Type in a search term (i.e. one of your keywords) and see what clusters are returned.



Corpus Files
Chipotle.txt
Honda.txt
JetBlue.txt
Netflix.txt
PapaJohns.txt
Samsung.txt
Starbucks.txt
Tesla.txt
Toyota.txt
UnitedAirlines.txt
WellsFargo.txt

Total No. of Cluster Types 83 Total No. of Cluster Tokens 171

Rank	Freq	Range	Cluster
1	57	9	i
2	8	4	i am
3	5	2	i want
4	4	2	i want to
5	3	2	i am deeply
6	3	1	i extend
7	3	2	i know
8	3	2	i'm
9	3	2	i've
10	3	2	i've been
11	2	2	i also
12	2	1	i am accountable
13	2	2	i can
14	2	1	i extend my
15	2	2	i have
16	2	2	i'm sorry
17	1	1	i also love
18	1	1	i also offer
19	1	1	i am committed
20	1	1	i am going
21	1	1	i am, as
22	1	1	i apologize
23	1	1	i apologize to
24	1	1	i believe
25	1	1	i believe that
26	1	1	i can to
27	1	1	i can watch
28	1	1	i do
29	1	1	i do, because
30	1	1	i engage
31	1	1	i engage in
32	1	1	i extend the
33	1	1	i fear
34	1	1	i fear the
35	1	1	i feel

Search Term ☑ Words □ Case □ Regex □ N-Grams
i
Cluster Size Min. 1 Max. 3
Min. Freq. 1 Min. Range 1
Sort by □ Invert Order
Sort by Freq
Search Term Position ☑ On Left □ On Right

Total No.
11
Files Processed

Figure 4.17 Analysis of a study corpus with clusters.

Memo 4.2: Emerging Patterns

Reflect on keywords that might be emerging from the data as you examine the results in AntConc. Consider what keywords might be worth deeper investigation. Consider also what these emerging keywords might mean in terms of your research questions.

Using Automated Coding

With manual coding the researcher is able to pay closer attention to details in the language and perhaps apply codes that require more interpretation. When using automated techniques, the researcher can more easily code greater quantities of data, even approaching the ability to code some corpora of data at scale (see Procedure 4.6, Excel Procedure 4.7, and MAXQDA Procedure 4.8). For example, imagine attempting to code a data set of interactions from student team meetings. If we are talking about data from a single semester, across a handful of classes, that is a manageable data set for manual coding. But imagine a larger data set that includes all student teams across all sections of a class across multiple years. Imagine now including student teams from different schools. The data set quickly gets out of hand as a project for the manual coder and heightens the appeal of tools that can be used for automating the coding, at least as a first pass in the analysis. In Figure 4.19, for example, we have an example where the clauses shown in Column D have been automatically coded for the presence or absence of various modals such as may, might, can, could, will, and would.

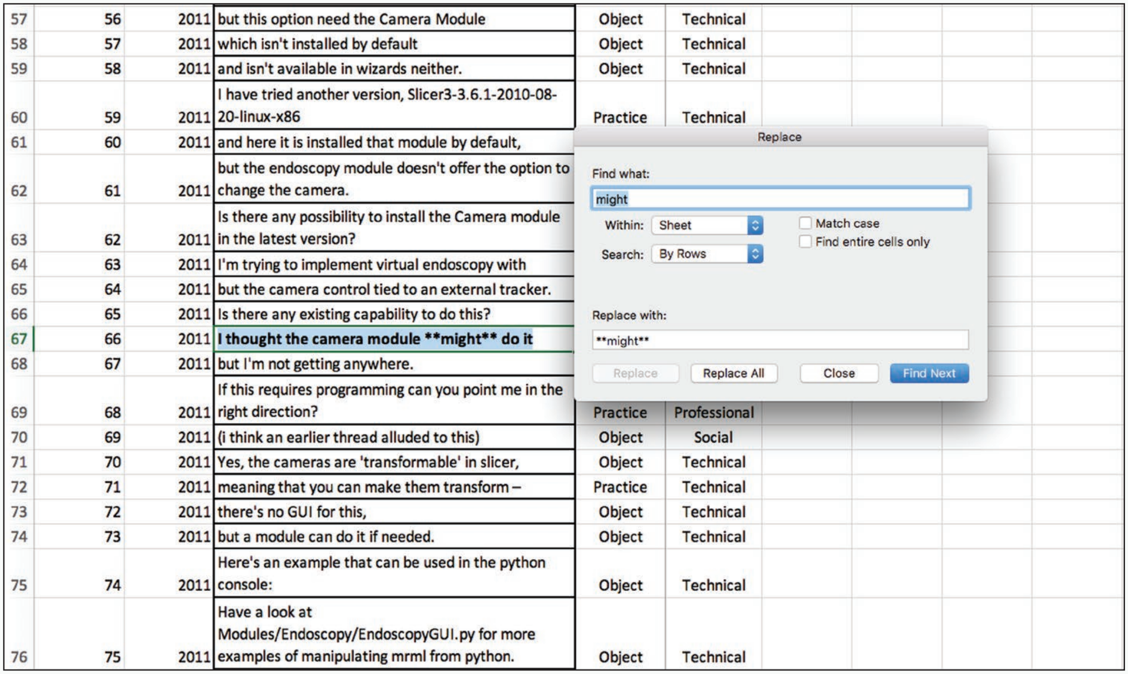
Correcting Autocoding

Procedures for automated coding will often overcode. That is, they will assign code to segments to which the code should not properly be applied. For this reason, most autocoding procedures require some kind of correction process. The easiest, though not quickest, way to correct autocoding is to inspect the coded data one segment at a time and remove any inappropriately applied codes. Often, the judicious use of wildcards or spaces in the search string can eliminate inappropriate matches. In any case, when setting up a new automated coding procedure, make sure to inspect the results of a small selection of data to insure you are getting what you intend.

Procedure 4.6: Automated Coding in Word

https://goo.gl/5Q4Jgr

1. Select the **Find** command under the **Edit** menu bar.
2. Select the the **Replace ...** option.



57	56	2011	but this option need the Camera Module	Object	Technical
58	57	2011	which isn't installed by default	Object	Technical
59	58	2011	and isn't available in wizards neither.	Object	Technical
60	59	2011	I have tried another version, Slicer3-3.6.1-2010-08-20-linux-x86	Practice	Technical
61	60	2011	and here it is installed that module by default,		
62	61	2011	but the endoscopy module doesn't offer the option to change the camera.		
63	62	2011	Is there any possibility to install the Camera module in the latest version?		
64	63	2011	I'm trying to implement virtual endoscopy with		
65	64	2011	but the camera control tied to an external tracker.		
66	65	2011	Is there any existing capability to do this?		
67	66	2011	**I thought the camera module **might** do it**		
68	67	2011	but I'm not getting anywhere.		
69	68	2011	If this requires programming can you point me in the right direction?	Practice	Professional
70	69	2011	(i think an earlier thread alluded to this)	Object	Social
71	70	2011	Yes, the cameras are 'transformable' in slicer,	Object	Technical
72	71	2011	meaning that you can make them transform –	Practice	Technical
73	72	2011	there's no GUI for this,	Object	Technical
74	73	2011	but a module can do it if needed.	Object	Technical
75	74	2011	Here's an example that can be used in the python console:	Object	Technical
76	75	2011	Have a look at Modules/Endoscopy/EndoscopyGUI.py for more examples of manipulating mrml from python.	Object	Technical

Replace

Find what: might

Within: Sheet

Search: By Rows

□ Match case

□ Find entire cells only

Replace with: **might**

Figure 4.18: Using find and replace to highlight key words.

3. As shown in Figure 4.18, type the specific word or phrase to be automatically coded (such as might) into the **Find** field and something easier to find (such as **might**) in the **Replace** field.
4. Click **Replace All**.

Not only will Word make all the replacements, but it will also report to you how many such replacements it made.

Excel Procedure 4.7: Automated Coding in Excel

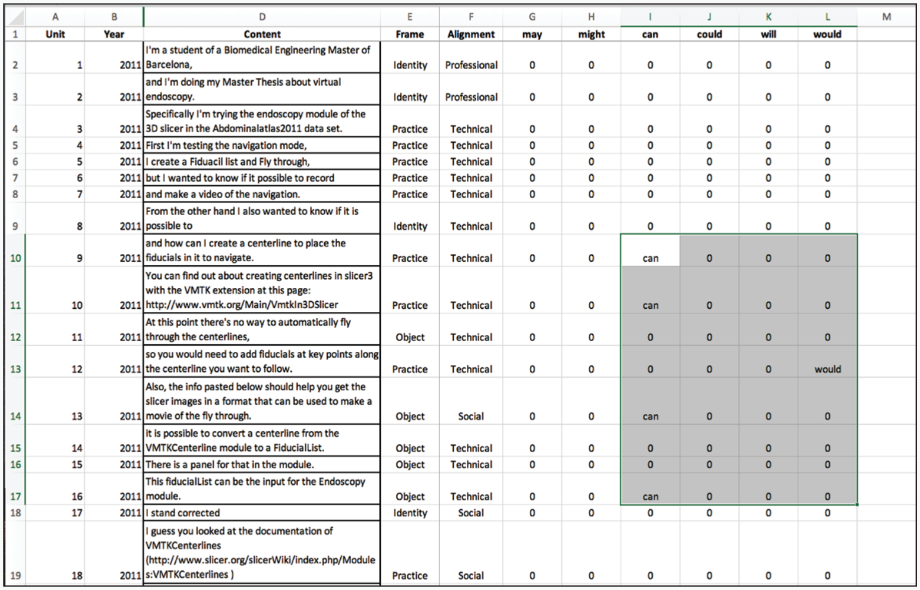
https://goo.gl/5Q4Jgr

Automated pattern matching in Excel is accomplished as follows:

1. Into Row 1, type the keyword or phrase you want to code for.
2. Into the first data cell of your coding column, type the following formula:

   =IF(ISERR(SEARCH(PATTERN,SEGMENT)),0,PATTERN)

3. Edit the formula as follows:
   - Replace PATTERN with the name of the coding column followed by $1.
   - Replace SEGMENT with the name of the cell in which the first data segment is found preceded by a $.
4. Drag the formula down to fill the rest of the column until the last line of data.



	A	B	D	E	F	G	H	I	J	K	L	M
1	Unit	Year	Content	Frame	Alignment	may	might	can	could	will	would	
2	1	2011	I'm a student of a Biomedical Engineering Master of Barcelona,	Identity	Professional	0	0	0	0	0	0	
3	2	2011	and I'm doing my Master Thesis about virtual endoscopy.	Identity	Professional	0	0	0	0	0	0	
4	3	2011	Specifically I'm trying the endoscopy module of the 3D slicer in the Abdominalatlas2011 data set.	Practice	Technical	0	0	0	0	0	0	
5	4	2011	First I'm testing the navigation mode,	Practice	Technical	0	0	0	0	0	0	
6	5	2011	I create a Fiduacil list and Fly through,	Practice	Technical	0	0	0	0	0	0	
7	6	2011	but I wanted to know if it possible to record	Practice	Technical	0	0	0	0	0	0	
8	7	2011	and make a video of the navigation.	Practice	Technical	0	0	0	0	0	0	
9	8	2011	From the other hand I also wanted to know if it is possible to	Identity	Technical	0	0	0	0	0	0	
10	9	2011	and how can I create a centerline to place the fiducials in it to navigate.	Practice	Technical	0	0	can	0	0	0	
11	10	2011	You can find out about creating centerlines in slicer3 with the VMTK extension at this page: http://www.vmtk.org/Main/VmtkIn3DSlicer	Practice	Technical	0	0	can	0	0	0	
12	11	2011	At this point there's no way to automatically fly through the centerlines,	Object	Technical	0	0	0	0	0	0	
13	12	2011	so you would need to add fiducials at key points along the centerline you want to follow.	Practice	Technical	0	0	0	0	0	would	
14	13	2011	Also, the info pasted below should help you get the slicer images in a format that can be used to make a movie of the fly through.	Object	Social	0	0	can	0	0	0	
15	14	2011	it is possible to convert a centerline from the VMTKCenterline module to a FiducialList.	Object	Technical	0	0	0	0	0	0	
16	15	2011	There is a panel for that in the module.	Object	Technical	0	0	0	0	0	0	
17	16	2011	This fiducialList can be the input for the Endoscopy module.	Object	Technical	0	0	can	0	0	0	
18	17	2011	I stand corrected	Identity	Social	0	0	0	0	0	0	
19	18	2011	I guess you looked at the documentation of VMTKCenterlines (http://www.slicer.org/slicerWiki/index.php/Modules:VMTKCenterlines)	Practice	Social	0	0	0	0	0	0	

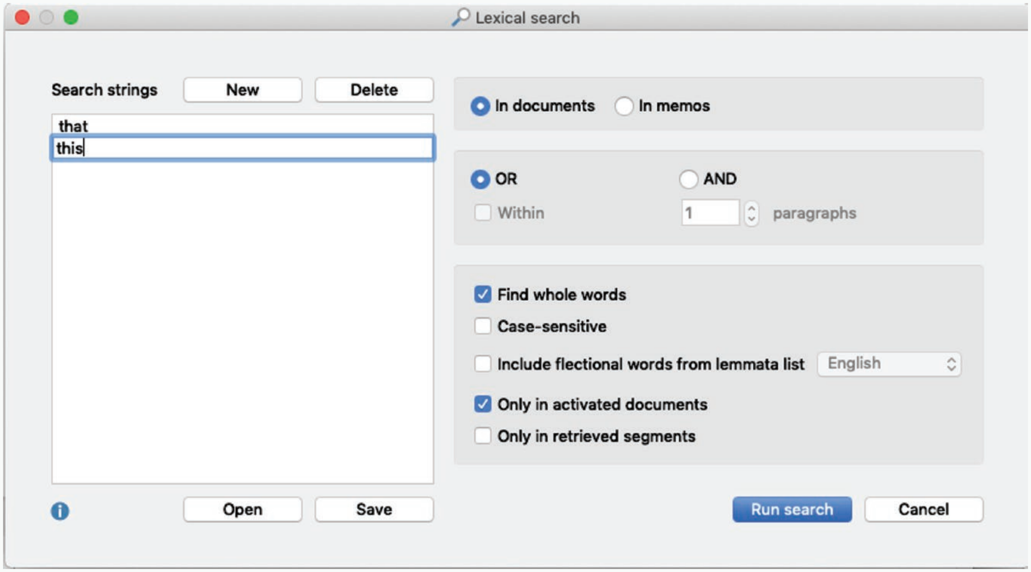
Figure 4.19: Automated coding in Excel.

Note that, in Figure 4.19, we have an example where the clauses shown in Column D have been automatically coded for the presence or absence of various modals such as may, might, can, could, will, and would.

MAXQDA Procedure 4.8: Automated Coding in MAXQDA

https://goo.gl/5Q4Jgr

MAXQDA has a variety of procedures for automated coding which can be accessed under **Lexical Search** on the **Analysis** menu. To autocode in MAXQDA as shown in Figure 4.20:


Lexical search

Search strings
that
this

● In documents ○ In memos

● OR ○ AND
□ Within 1 paragraphs

☑ Find whole words
□ Case-sensitive
□ Include flectional words from lemmata list English
☑ Only in activated documents
□ Only in retrieved segments

Figure 4.20: Autocoding with lexical search in MAXQDA.

1. Select **Lexical Search** from the **Analysis** menu.
2. Add keywords by clicking **New** and typing in one or more keywords for which you want to search.
3. Choose to search in documents and find whole words.
4. Click **Run Search**.
5. In the **Search Results** window, click on the **Autocode search results** icon and then chose the code with which you want to autocode.
6. Chose **paragraph** as the unit for autocoding and click **autocode**.
7. Inspect autocoded segments by clicking on the segment in the **Search Results** window.

The segment will appear in the **Document Browser** window.

Exercise 4.2 Try It Out

In one of your data files, use automated coding for the word “we” in either Excel or MAXQDA. Inspect and describe the results. Which of these are appropriate? Which of these are plainly wrong. Devise one or two methods to reduce the errors.

For Discussion: Under what circumstances could you imagine using automatic coding with your data? What is it good for? In what ways is it limited?

Nested Coding

A complex coding situation can arise when you want to nest one coding scheme within another. Suppose you code the conversational turns of a tutor in a writing center using a coding scheme that includes *responds to text*, *discusses assignment*, and *talks about other things*. You then want to go on to look more closely at those turns that *responds to text*, to decide whether they were facilitative or directive. You would then be using nested coding schemes in which the second dimension (*facilitative* vs. *directive*) was applied selectively only to data that had been placed in a specific category as a result of the first coding scheme.

Nested coding can also be used following automated coding (see Excel Procedure 4.8 and MAXQDA Procedure 4.9). The automated coding produces a selection of segments that the second or nested coding can further analyze. Such a nesting procedure not only allows you to focus on the results of the automated coding but also allows you to correct for any overcoding that may have occurred.

Enumerative Coding Schemes

The kind of coding schemes we have been talking about so far in this chapter can be thought of as procedural. They provide decision rules that will allow us to place each segment into the category intended by the researcher.

Excel Procedure 4.8: Nested Coding in Excel

https://goo.gl/5Q4Jgr

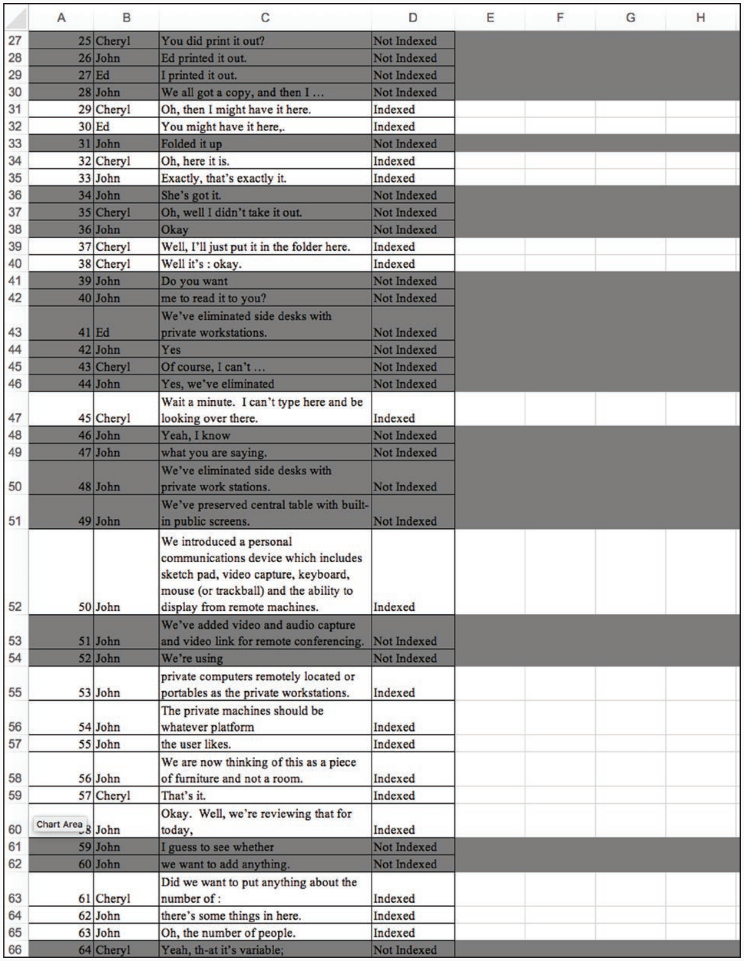
In Excel, to prepare for nested coding of data once the first coding is complete,

1. Grey out the cells for which no further coding is needed as shown in Figure 4.21.

Here, we are preparing to further code just those segments that were coded as “Indexed” using the first coding scheme in Column D.

2. Use the next column for your second nested coding as shown in Column E.

The dark shading in cells in Column E tells us that those particular segments are not to be coded, but by keeping them in the worksheet while we code, we have access to the full context of the surrounding segments.



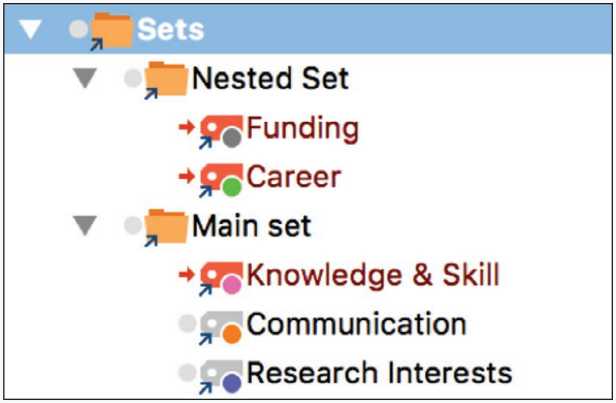
	A	B	C	D	E	F	G	H
27	25	Cheryl	You did print it out?	Not Indexed				
28	26	John	Ed printed it out.	Not Indexed				
29	27	Ed	I printed it out.	Not Indexed				
30	28	John	We all got a copy, and then I …	Not Indexed				
31	29	Cheryl	Oh, then I might have it here.	Indexed				
32	30	Ed	You might have it here,.	Indexed				
33	31	John	Folded it up	Not Indexed				
34	32	Cheryl	Oh, here it is.	Indexed				
35	33	John	Exactly, that’s exactly it.	Indexed				
36	34	John	She’s got it.	Not Indexed				
37	35	Cheryl	Oh, well I didn’t take it out.	Not Indexed				
38	36	John	Okay	Not Indexed				
39	37	Cheryl	Well, I’ll just put it in the folder here.	Indexed				
40	38	Cheryl	Well it’s : okay.	Indexed				
41	39	John	Do you want	Not Indexed				
42	40	John	me to read it to you?	Not Indexed				
43	41	Ed	We’ve eliminated side desks with private workstations.	Not Indexed				
44	42	John	Yes	Not Indexed				
45	43	Cheryl	Of course, I can’t …	Not Indexed				
46	44	John	Yes, we’ve eliminated	Not Indexed				
47	45	Cheryl	Wait a minute. I can’t type here and be looking over there.	Indexed				
48	46	John	Yeah, I know	Not Indexed				
49	47	John	what you are saying.	Not Indexed				
50	48	John	We’ve eliminated side desks with private work stations.	Not Indexed				
51	49	John	We’ve preserved central table with built-in public screens.	Not Indexed				
52	50	John	We introduced a personal communications device which includes sketch pad, video capture, keyboard, mouse (or trackball) and the ability to display from remote machines.	Indexed				
53	51	John	We’ve added video and audio capture and video link for remote conferencing.	Not Indexed				
54	52	John	We’re using	Not Indexed				
55	53	John	private computers remotely located or portables as the private workstations.	Indexed				
56	54	John	The private machines should be whatever platform	Indexed				
57	55	John	the user likes.	Indexed				
58	56	John	We are now thinking of this as a piece of furniture and not a room.	Indexed				
59	57	Cheryl	That’s it.	Indexed				
60	Chart Area 8	John	Okay. Well, we’re reviewing that for today,	Indexed				
61	59	John	I guess to see whether	Not Indexed				
62	60	John	we want to add anything.	Not Indexed				
63	61	Cheryl	Did we want to put anything about the number of :	Indexed				
64	62	John	there’s some things in here.	Indexed				
65	63	John	Oh, the number of people.	Indexed				
66	64	Cheryl	Yeah, th-at it’s variable;	Not Indexed				

Figure 4.21: Preparing for nested coding in Excel.

MAXQDA Procedure 4.9: Nested Coding in MAXQDA

https://goo.gl/5Q4Jgr

In MAXQDA, you can create nested coding using separate code sets for the main and nested coding as shown in Figure 4.22.

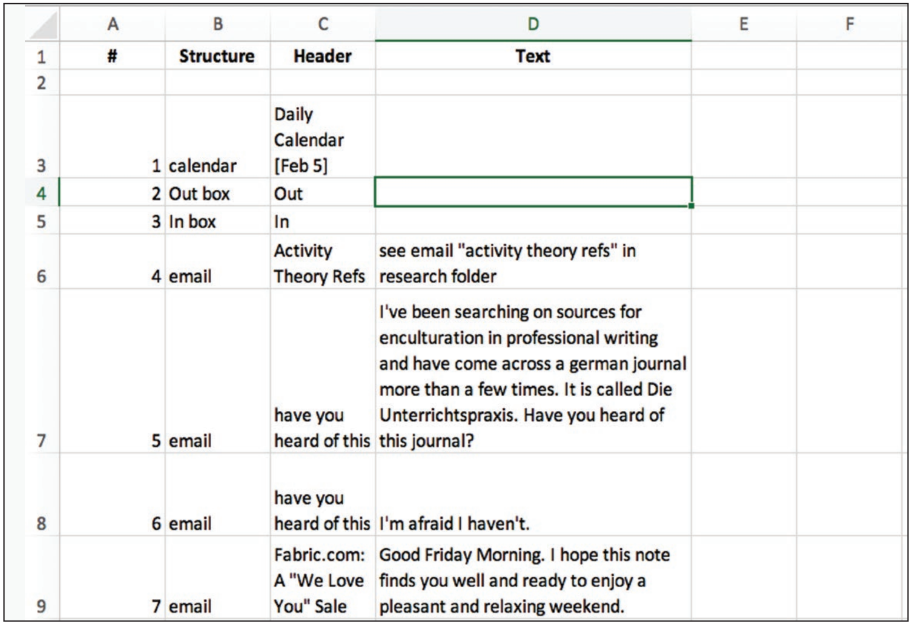


Sets
Nested Set
Funding
Career
Main set
Knowledge & Skill
Communication
Research Interests

Figure 4.22: Main and nested coding sets in MAXQDA.

1. In the **Code System** window, right click on **Sets** and choose the **New Set** command.
2. Type in the name of your main dimension.
3. Drag and drop into the main coding set appropriate codes from your **Code System** list.
4. Right click in the grey area of the coding column in the **Document Browser** and choose the option **Only activated codes**.
5. Activate the main coding set.
6. Code your data with the main coding set.
7. To create your nested dimension, create a nested coding set by right clicking on **Sets** and choosing the **New Set** command.
8. Drag and drop into the nested coding set appropriate codes from your **Code System** list.
9. To code with the nested coding scheme only those segments that were coded with a specific code in the main dimension, activate just this specific code in the main coding set.
10. Use the codes in the nested code set to code the segments with visible codes (from the main coding set) in the **Document Browser**.

A second kind of coding scheme exists that is enumerative rather than procedural. Instead of providing cases and examples of those cases, a complete enumeration is provided. In Figure 4.23, for example, a complete enumeration is provided of the text codes used in a study of desktop activity. Each distinct text that was accessed during the desktop session is listed here and assigned its own number. In the data itself then, these numeric codes have been used to code the data for the dimension of Text.



	A	B	C	D	E	F
1	#	Structure	Header	Text		
2						
3	1	calendar	Daily Calendar [Feb 5]			
4	2	Out box	Out			
5	3	In box	In			
6	4	email	Activity Theory Refs	see email "activity theory refs" in research folder		
7	5	email	have you heard of this	I've been searching on sources for enculturation in professional writing and have come across a german journal more than a few times. It is called Die Unterrichtspraxis. Have you heard of this journal?		
8	6	email	have you heard of this	I'm afraid I haven't.		
9	7	email	Fabric.com: A "We Love You" Sale	Good Friday Morning. I hope this note finds you well and ready to enjoy a pleasant and relaxing weekend.		

Figure 4.23: Enumerative coding.

Generally, a procedural coding scheme is to be preferred to an enumerative scheme because neither your second coders nor your readers can hold in their minds and make meaning of long lists. A few important exceptions exist. First, if the dimension shows relatively small variation along recognizable categories, an enumerative list can suffice. Second, if the concept underlying the dimension is hard to grasp, perhaps because the distinction is part of the culture being studied but not part of the culture of your second coder or your readers, an enumeration may be the best way to communicate the fuzzy set.

Memo 4.3: Coding Scheme Rationale

Examine the coding scheme you have developed. Explain the choices behind it:

- What kind of coding scheme is it?
- How is it related to prior research?
- How does it address your research question?

Selected Studies Using Procedural Coding

Angeli, E. L. (2015). Three types of memory in emergency medical services communication. *Written Communication, 32*(1), 3-38.

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. https://doi.org/10.1177/0741088316668517

Geisler, C., Rogers, E. H., & Haller, C. R. (1998). Disciplining discourse: Discourse practice in the affiliated professions of software engineering design. *Written Communication, 15*(1), 3-24. https://doi.org/10.1177/0741088398015001001

Jones, J. (2008). Patterns of revision in online writing: A study of Wikipedia's featured articles. *Written Communication, 25*(2), 262-289.

Lancaster, Z. (2014). Exploring valued patterns of stance in upper-level student writing in the disciplines. *Written Communication, 31*(1), 27-57. https://doi.org/10.1177/0741088313515170

Perl, S. (1979). The composing processes of unskilled college writers. *Research in the Teaching of English, 13*(4), 317-336.

Selected Studies Using Automated Coding

Abbasi, A. (2007). Affect intensity analysis of Dark Web forums. *IEEE Intelligence and Security Informatics* (pp. 282-288). https://doi.org/10.1109/ISI.2007.379486

Steinfeld, N., & Lev-On, A. (2014). "Well-done, Mr. Mayor!": Linguistic analysis of municipal Facebook pages. *Proceedings of the 15th Annual International Conference on Digital Government Research*, USA, 273-279. https://doi.org/10.1145/2612733.2612763

For Further Reading

Bazerman, C. (1988). *Shaping written knowledge. The genre and activity of the experimental article in science*. Madison, WI: University of Wisconsin Press.

Boettger, R. K., & Palmer, L. A. (2010). Quantitative content analysis: Its use in technical communication. *IEEE Transactions on Professional Communication, 53(4),* 346-357. https://doi.org/10.1109/TPC.2010.2077450

Bourque, L. B. (2004). Coding. In M. S. Lewis-Beck, A. Bryman, & T. Futing Liao (Eds.), *The Sage encyclopedia of social science research methods* (132-136). Thousand Oaks, CA: Sage.

Coffey, A., & Atkinson, P. (1996). Concepts and coding. In A. Coffey & B. Atkinson (Eds.), *Making sense of qualitative data: Complementary research strategies* (pp. 26-53). Thousand Oaks, CA: Sage.

Gee, J. P. (2005). *An introduction to discourse analysis: Theory and method.* New York: Routledge.

Kelle, U. (2000). Computer-assisted analysis: Coding and indexing. In M. W. Bauer & G. Gaskell (Eds.), *Qualitative researching with text, image, and sound* (pp. 282-298). Thousand Oaks, CA: Sage.

Krippendorff, K. (2013). *Content analysis: An introduction to its methodology* (3rd ed.). Los Angeles: Sage.

Kronberger, N., & Wagner, W. (2000). Keywords in context: Statistical analysis of text features. In M. W. Bauer & G. Gaskell (Eds.), *Qualitative researching with text, image, and sound* (pp. 299-317). Thousand Oaks, CA: Sage.

Miles, M. B., & Huberman, A. M. (1994). Codes and coding. In M. B. Miles, A. M. Huberman, & J. Saldaña (Eds.), *Qualitative data analysis: An expanded sourcebook* (pp. 55-65). Thousand Oaks, CA: Sage.

Neuendorf, K. (2016). *The content analysis guidebook*. London: Sage Publications.

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

Scott, M. (1997). PC analysis of key words—and key key words. *System*, *25*(2), 233-245.

Stemler, S. (2001). An overview of content analysis. *Practical Assessment, Research & Evaluation, 7*(17). Retrieved from http://pareonline.net/getvn.asp?v=7&n=17

Wenger, E. (1998). *Communities of practice: Learning, meaning, and identity*. Cambridge, UK: Cambridge University Press.