Qualitatiative data analysis

Qualitative data recap

- Videos, audios, transcripts (from interviews, focus groups)
- Observation notes
- Images (e.g., photographs of artifacts)
- Reviews / feedback forms / open ended answers in surveys
- Documents
- Diary entries

- Formats → text, images, audio, video
- Nature: Less structure, high variability (each one uses different words for same concept), lots of detail, subjective experiences & opinions

Qualitative data analysis

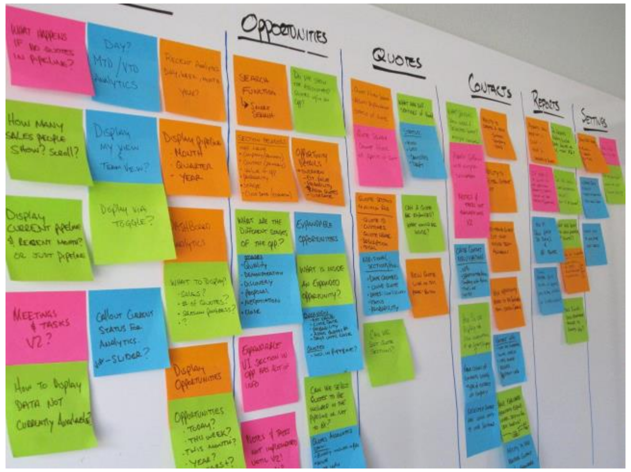
- What kinds of research questions?
  - What, how, why => detailed and rich
  - Mostly, qualitative; can do some quantitative, but not reliable
- We know how to gather qualitative data & their characteristics?
- Purpose of analysis:
  - Do “whatever is needed” with the qualitative data, to answer research questions
- Typically, qualitative data analysis happens via “qualitative coding”, “thematic analysis” or “card sorting”

Card sorting

- Take qualitative data, each piece of data in a “card”
- Sort them according to topics / under headings (see picture)

- How to choose headings?
  - Predefined headings (from questions, prior research, etc)
  - Put related things together, and add a heading later

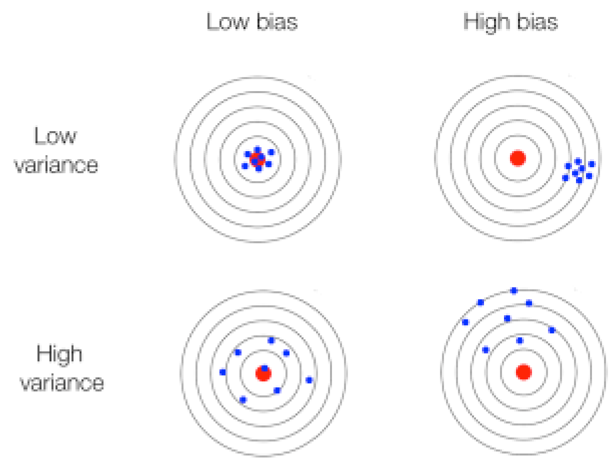
- When is it useful?
  - Organize content under topics (website menu design)
  - See what stack is big
    - positive / negative / neutral feedback
    - Complaints (customer service, payments, delivery, etc.)



Card sorting

- Great for one kind of data, that needs to be put in categories
- Great for small number of categories
- Great when one data goes under only one category (mutually exclusive)
- Once we have a pile under category:
  - Which category is biggest (and needs most attention)
  - Sub-categorize selected ones for further insights

Errors and biases



- Error: random (high variance and deviation),
  - Evens out with a lot of data
  - E.g., “sad people might be critical”, but “happy people should be overly positive” thus balancing out, in a large sample.

- Bias: systematic shift in data (e.g., all say +ve, for social reasons)
  - Doesn’t go off with gathering a lot of data

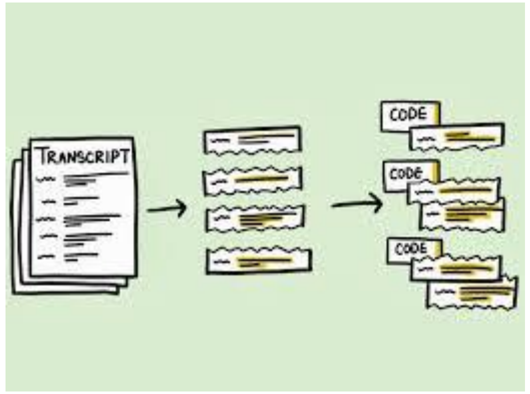
- As we said before, qualitative data has high variance
  - Say the same thing, but in a variety of ways

Errors and bias

- Researchers' go off with large sample, and triangulation
  - e.g., confirm card sorting with a large survey
- Biases:
  - Researchers systematically sort cards in wrong places (e.g., this has too many options – one might think this is a good thing!)
  - How to minimize:
    - Have more than one person card sort
    - Simultaneous multiple sorters: allow debate and consensus => take a card, debate, agree where to put, if not, debate and come to agreement. [Sometimes, debate is useful; helps see thought process, and even put something under two categories]
    - Separate sessions for each sorter, take averages / commonalities. Find other ways to deal with disagreements (e.g., good/bad – look for more data, go back and ask, etc.)
    - Remember, this is called participant / researcher triangulation.

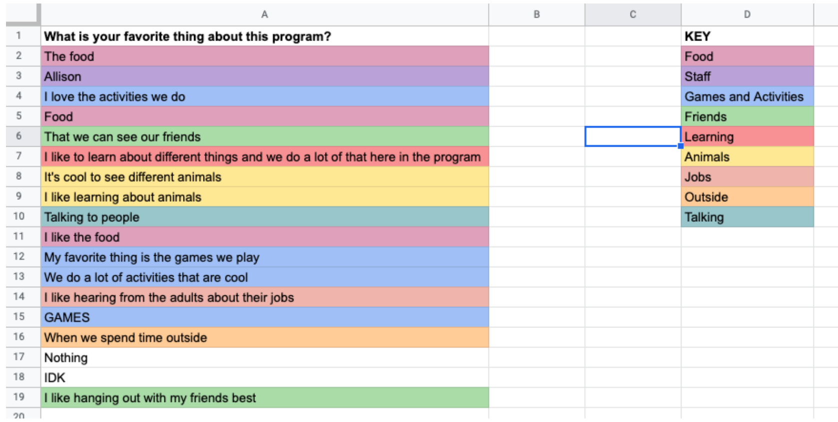
Qualitative coding

- Assign “labels” to content
- Used when content is not homogenous (e.g., interview transcripts with various answers to various questions)
- Multi-step: Unitization → Coding → Thematic analaysis



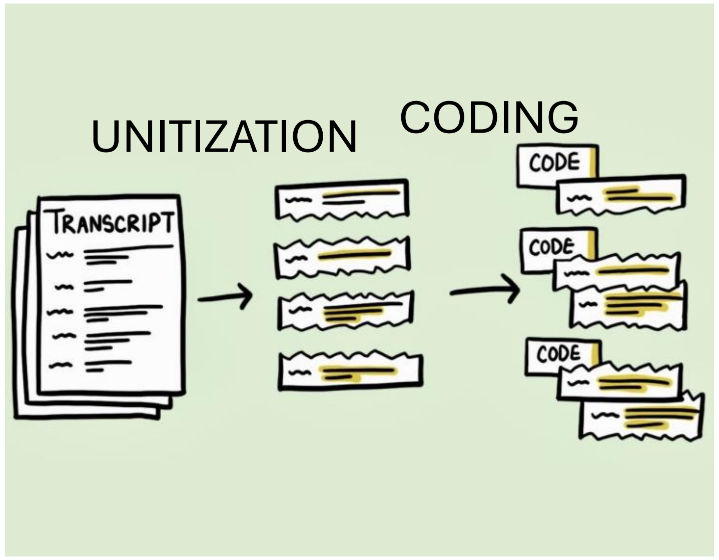
- First, unitization
  - Break down “unstructured” content into “units” to be coded
  - Interval Segmentation (every 10/20/30s, every minute, etc.)
  - Topic-based (every paragraph, every time participant mentions a new instance or topic, every question, etc.)
    - Subjective & error/bias prone; so, one researcher breaks down, another reviews.
  - Every review, every participant response to open ended survey question, every document, etc.

Coding



	A	B	C	D
1	What is your favorite thing about this program?			KEY
2	The food			Food
3	Allison			Staff
4	I love the activities we do			Games and Activities
5	Food			Friends
6	That we can see our friends			Learning
7	I like to learn about different things and we do a lot of that here in the program			Animals
8	It's cool to see different animals			Jobs
9	I like learning about animals			Outside
10	Talking to people			Talking
11	I like the food			
12	My favorite thing is the games we play			
13	We do a lot of activities that are cool			
14	I like hearing from the adults about their jobs			
15	GAMES			
16	When we spend time outside			
17	Nothing			
18	IDK			
19	I like hanging out with my friends best			
20				

- Assign “codes” (labels) to each unit
- How to do it?
  - We need data in units (see unitization on previous slide)
  - We need a codebook (codes, with their definitions)
    - Pre-defined / existing / already available [closed coding]
    - To be created from themes in data [open coding]
  - Then assign the codes in codebook, to the data units

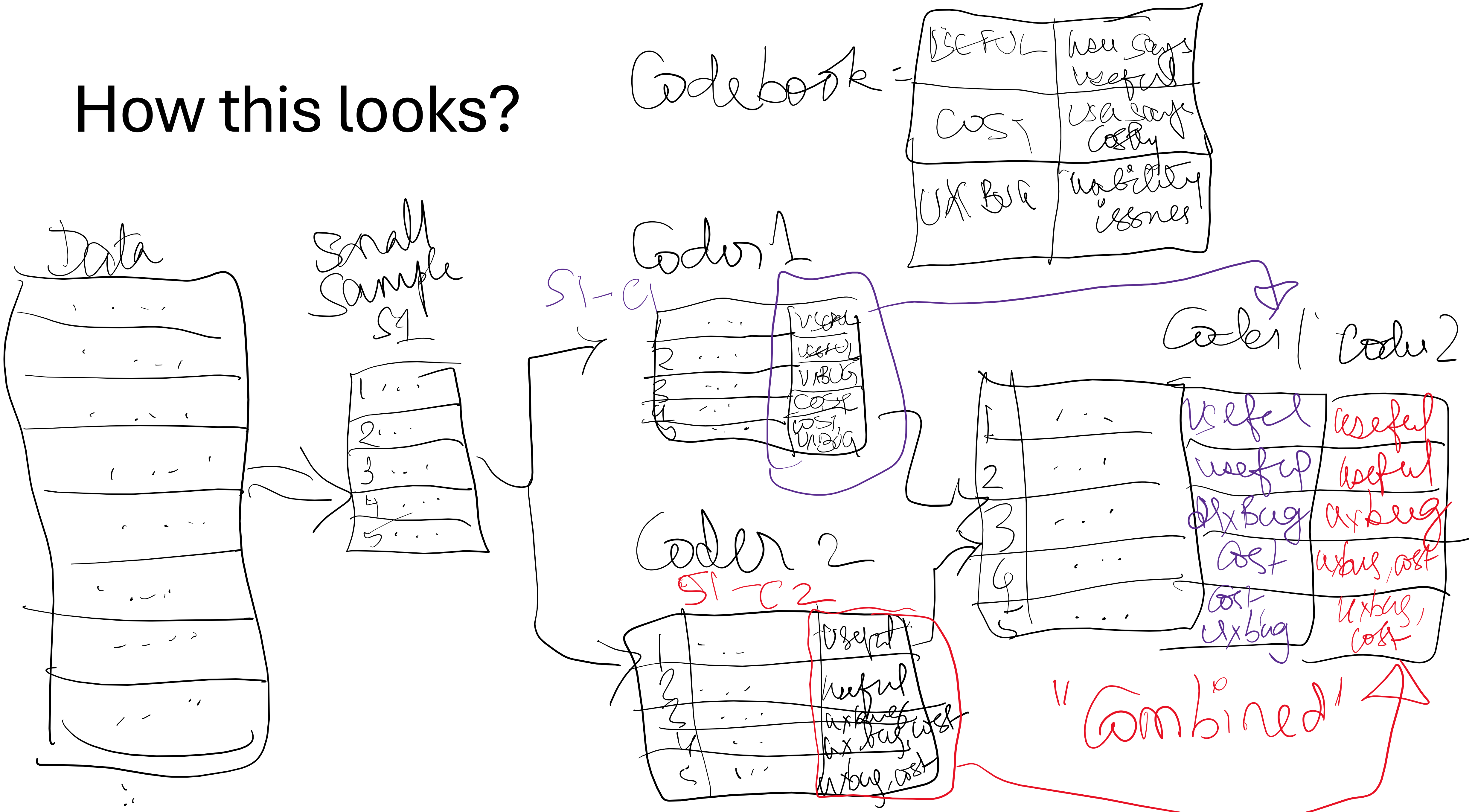


- Higher bias in interpreting data & assigning codes
- How to prevent?
  - More than one coder aka researcher triangulation

Coding in practice

- **Situation-1:** Code book already available (closed coding)
- Start by taking a small sample of unitized data, let's call it S1
    - 1-2% of total "units", taken from various parts of transcript, participants
- Decide on:
    - How many codes per segment/unit (e.g., major activity / every single activity)
    - What if an event is too small (say, only 2 sec in a 30 sec segment)?
    - What if an event goes across two segments (e.g., 20th second in first to 15th second in second).
    - Code category-wise (e.g., first code all positives, then all negatives, etc.)
- Two coders C1, C2 code S1 with the codebook, independently;
    - C1 codes S1 + codebook → S1-C1
    - C2 codes S1 + codebook → S1-C2
- Combine coded data, to compare the two coders' codes
    - S1-C1 + S2-C2 → S1, C1, C2

How this looks?



Inter-rater reliability (IRR)

- Now, see how the two coders agree with each other
  - IRR is a measure of agreement between coders
  - High agreement → less bias (two people independently think the same way) & coding process is repeatable.
  - Low agreement → bias to be ironed out, and/or codes need refinement

- How to computer IRR?
  - First combine coded sample from two coders, to make comparisons easy
    - (Saw this on previous slide)
  - Compute IRR measure for each data segment in Sample S1 (each row)
  - Average across all rows in sample S1

Inter-rater reliability (IRR)

- IRR measures → there are several
  - Simplest is Jaccard co-efficient.
- Suppose there is a segment s1 in the sample S1, and coders C1 and C2 assign a set of codes c1 and c2, respectively.
- Assuming multiple codes per segment, c1 and c2 are sets.
- Then,
  - Jaccard co-efficient = $N(c1 \cap c2) / N(c1 \cup c2)$

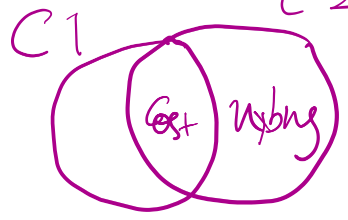
IRR computation

		Coder-1	Coder-2	IRR
1	...	useful	useful	1.0
2	...	useful	useful	1.0
3	...	ax bug	ax bug	1.0
4	...	cost	ax bug, cost	0.5
5	...	ax bug, cost	cost, ax bug	1.0
			Average	0.95

← agree on everything so 1.0 or 100%

" "

how?



$C1 \cap C2 = \{cost\}$

$C1 \cup C2 = \{cost, ax bug\}$

$n(C1 \cap C2) = 1;\ n(C1 \cup C2) = 2$

$\rightarrow 1/2 = 0.5$

← order changed, but still agree, so OK.

Note: Sometimes order matters; but not here!

95% agreement is great ← IRR

Inter-rater reliability (IRR)

- IRR should be high (80% or above)
- If IRR is low, then two coders discuss why they coded what they coded, revisit rules, code definitions, etc.
- With new rules, redo independent coding with new codebook and test if you agree on new sample S2.
- If so, revisit disagreements, but try a slightly bigger, new sample S3 (say 5% of data, then 10% of data and so on)
- When confident, take a fresh sample "S" making up 20% of data
- Agreement on this, with independent coding should be above 80%
- If not, go back and re-do!

Coding rest of data

- Once you 80% agreement on 20% data → save this!

- The remaining 80% of the data can be coded by only one of the two people
  - Code even if it has been used in previous samples for practice
    - (remember, this is the reals, and not trials!)
  - One coder could code all 80%, or could split up and code (each one codes 40% of the remaining data, or 50-30 % of the data, etc).
- What about agreement for this?
  - None, because there's only one rater, and we now know the two raters code more or less similarly (meaning, if the other one coded, it would be the same 80% of the time).

Sounds horrible?: negotiated agreement

- Sometimes, IRR with independent coding can be hard!
- Too many codes, something or the other is always missing, too much variation in data
- We use negotiated agreement
  - Two coders code 100% of data
  - Report agreement (often poor!)
  - The two coders look through each disagreement, and argue / negotiate out, and decide on how to code (both / one person wins!)
- Low reliability! (these two people had to argue, others would not agree either!).

Common issues

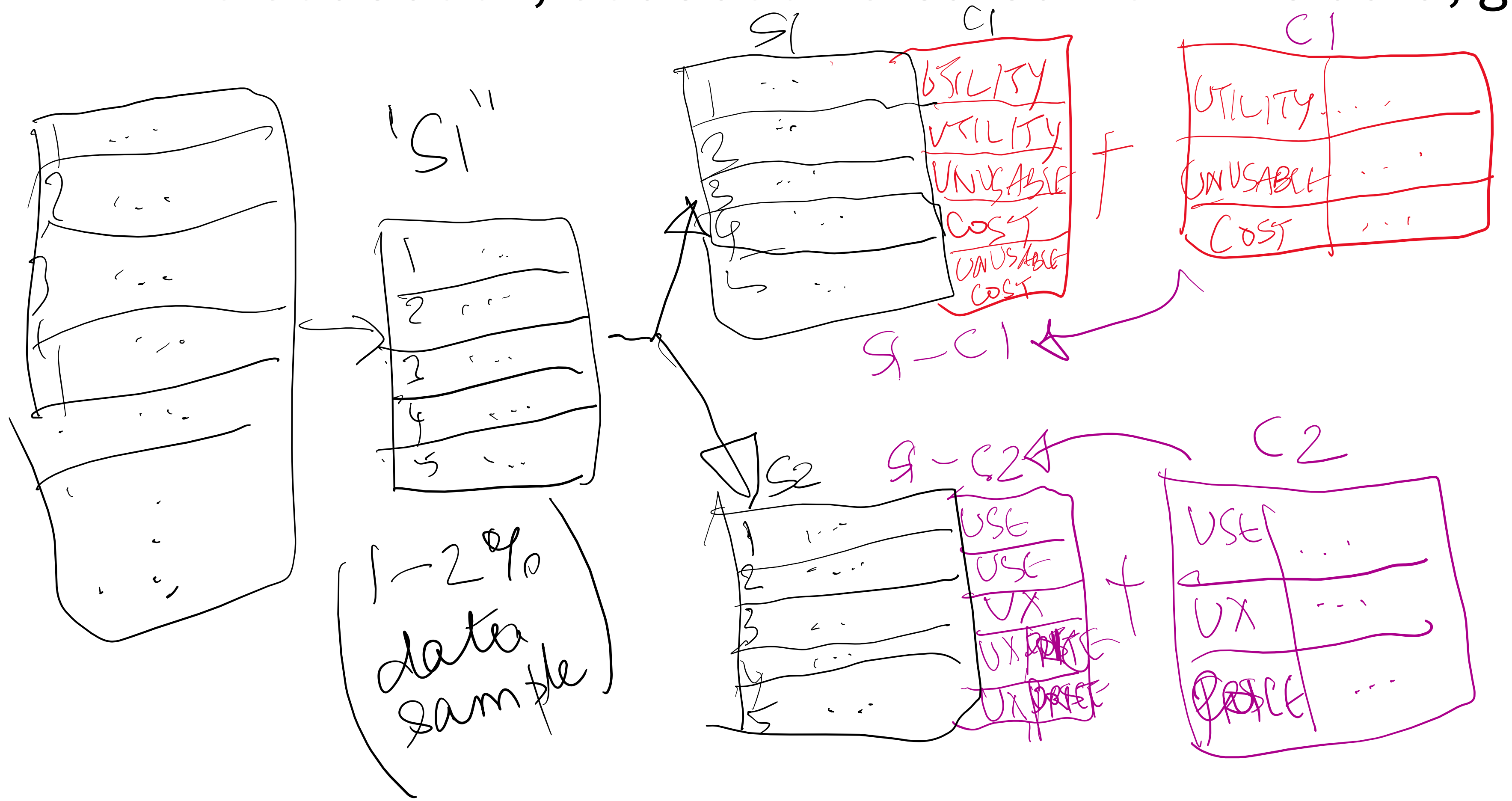
- Too many codes
  - Break it down to categories, code category wise
- There’s data, but not code (but you think code is important, or there’s a category with 4 items and you see a 5th one).
  - Add code to codebook, discuss as part of negotiation

- Every time codebook changes, use the newer codebook version (for both coders!)

- What if there is no prior codebook?
  - Open coding

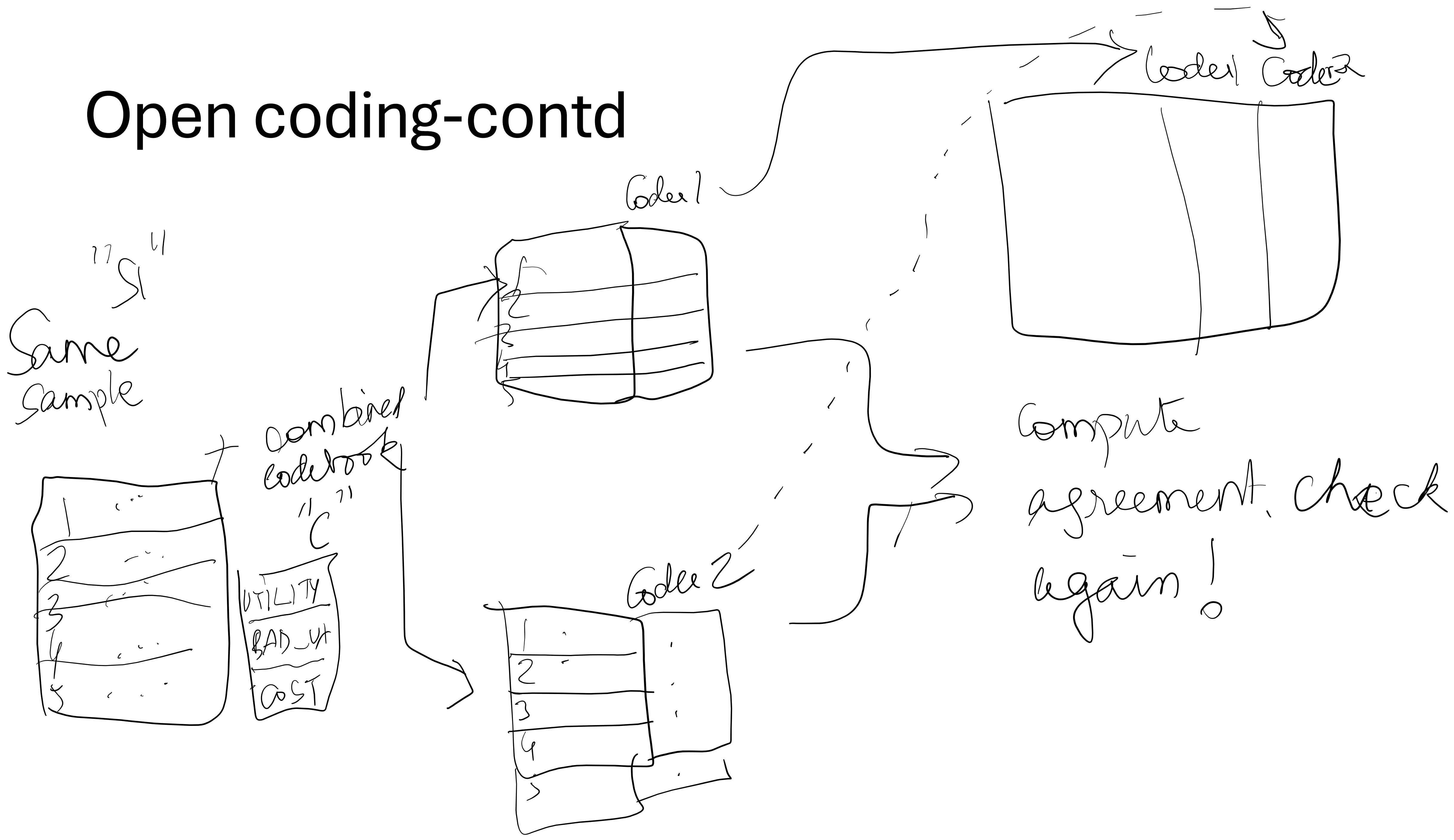
Open coding

- No codebook; codebook created from the data, ground up!



1. Same code, different names → give one name
2. Code in one, not in other → discuss and decide on one
3. Combine codes when needed, split codes when needed (one could use the same code, other could create a different code for the same thing!)

Open coding-contd

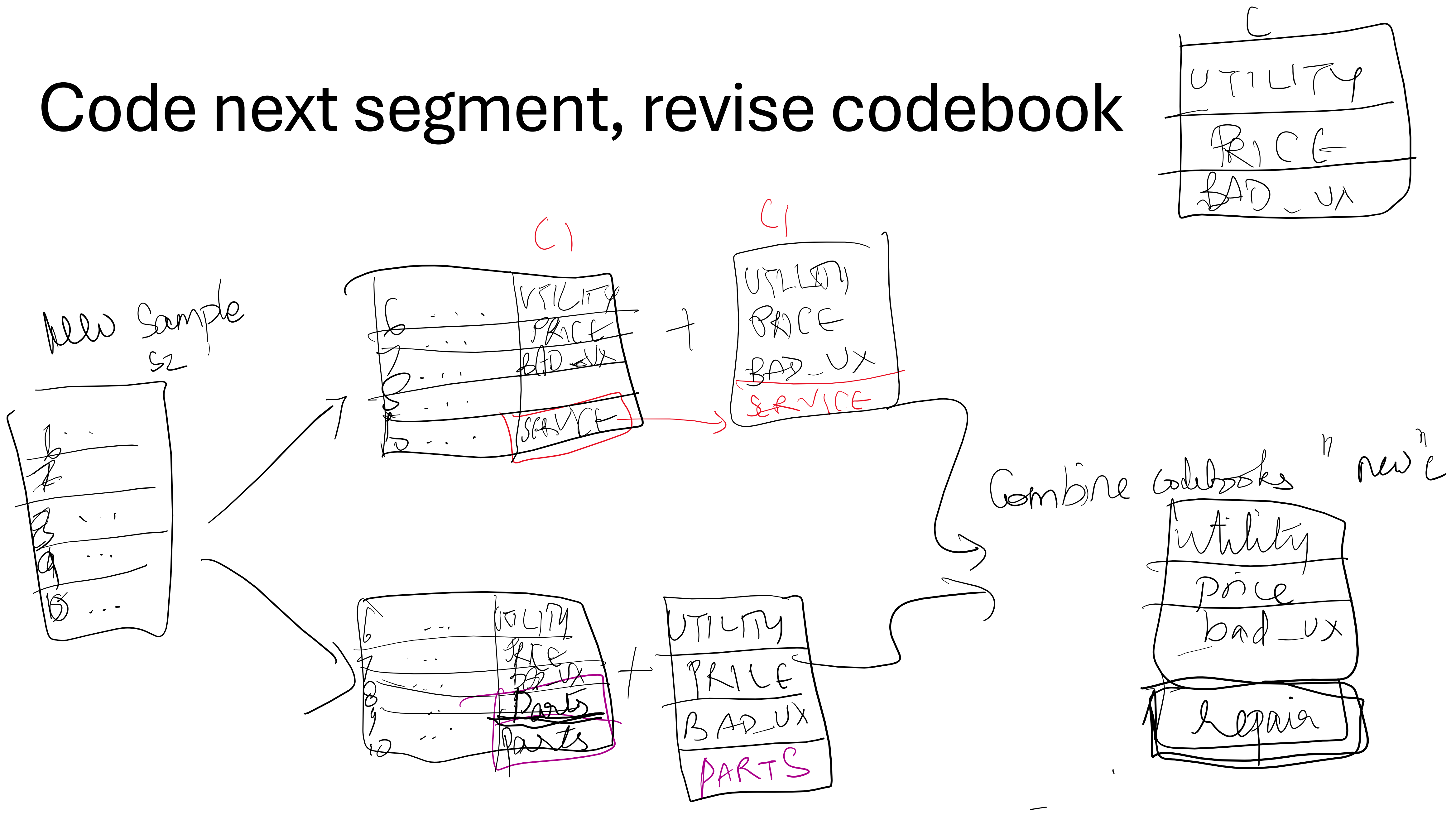


Why recode same segment?

- To see if, independently (without influencing each other), you still agree on codes.
- Yes – atleast these codes work!
- No – fix again!

- Note: IRR computed when same codebook is used, not otherwise!

Code next segment, revise codebook



Rinse and repeat

- Code S2 with “new C”
- If “new C” changed an old code, then even code S1 and S2 with new code
  - So you still agree on old data
  - Measure IRR and go on.
- When to stop?
  - No new codes are added, and IRR seems OK
- Pick new 20%, code independently with latest codebook and compare IRR; code the rest by one person.
- Alternatively, consider negotiated agreement

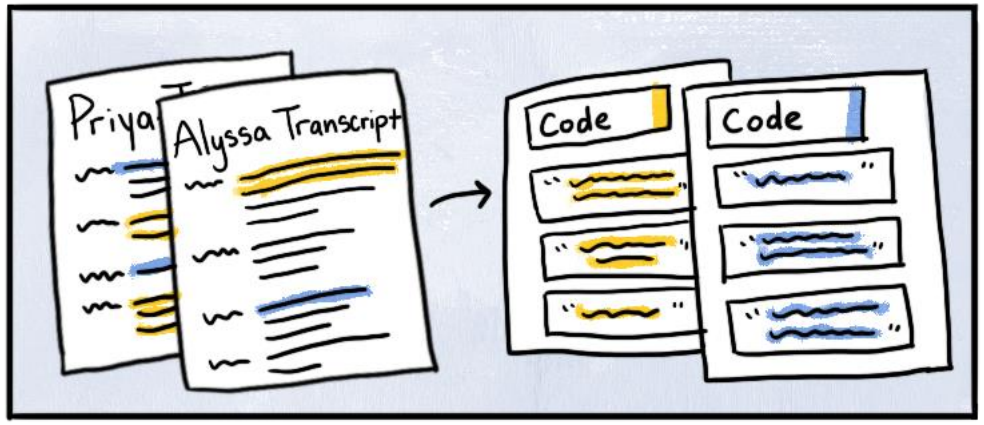
Analysis of “coded” data

- Measure frequency of codes (no. of times a code happened!)
  - Problem P and Q are the top issues, and account for 60% of all user complaints!
  - 60%, 18%, 12% , 10% of segments had activities A1, A2, A3 and A4 respectively --→ approximately indicate the time spent on each activity.
- Code co-occurrences (two codes come together in same segment)
  - E.g., Half the time (56%), code “ActivityA” occurred with code “ProblemP” ➔ about half the time, users face problem P, when doing activity A.
- Patterns → A always occurs preceded by B.
- State transitions → processes (A → B → C or D)

Thematic analysis

- Rigorous coding is hard, especially without codebook
- We use thematic coding!
  - Organize data into themes roughly
  - Identify “codebook” / themes by first reading through transcripts
  - Assign themes / codes to sections of data
  - Do as pair of researchers, or one does and other reviews

- Analyse each theme
  / relations between themes



Practice-1

What problems about the Reddit android app do users report on the Google play store?

Code what the review says (in-vivo coding)

What do people write in a review?

Code the kind of content in each review (structural coding)

- Interviewer-1 → Dataset-1 → Codebook 1
- Interviewer-2 → Dataset-1 → Codebook 2

- Discuss, combine to CodebookNew

- Interviewer-1 + CodebookNew + Dataset-1 → Coded Data-1
- Interviewer-2 + CodebookNew + Dataset-1 → Coded Data-2

- Compute agreement between Coded Data-1 and Coded Data-2

- Interviewer-1 + CodebookNew + Dataset-2 → Coded Data-21
- Interviewer-2 + CodebookNew + Dataset-2 → Coded Data-22

Practice-2: Coding your own interviews (homework-2)

Planning-1

- Re-read your RQs, code your data in ways to answer those questions.
  - Take a couple of minutes to independently skim through your interview notes to refresh memory
  - Think about what data answers what questions, and how you can code it.
  - Make notes for what you will code in the data, and under what categories

Planning-2

- Discuss with your partner, exchange notes
- Together, get a codebook started, and make notes for what categories go into that data.

Multiple interviewer notes

- There are two sets of transcripts for every participant.
- Please compare transcript notes for each participant
  - Are there any differences, in content?
  - A has something, B doesn’t have?
  - A and B mention two different things for the same phenomena?

- How can you go about coding such multiple notes?

Combining interview notes

- Option-1: Fresh copy combining both interviewers' notes; code the combined notes. Original notes remain as is!
- Option-2: Edit one interviewer's notes, to add other's notes; code the combined notes. Original notes change!
- Option-3: Each interviewer codes their own notes independently, and then compare codes, and look for disagreements and then negotiate. (Not great, but OK, sometimes!).

Do either Option-1 or Option-2 for homework!

Coding & analysis

- Qualitative coding, with less rigor in codebook creation!
- No independent codebook creation;
  - Take small samples, sit together to create codebook
  - Code independently, measure IRR, fix codebook together, repeat for new sample.
  - When codebook stable and IRR good, try on 20% data.

- Report IRR numbers, on independent coding of 20% data (should be atleast 80%)
- Code the rest of the data by one coder
- Analyze coded data, to answer your research questions

Readings

- Read Preece chapter-8 (until 8.4)