Custom GPT Workshop Handout

Designing Your Own Qualitative Analysis GPT

Kartik Trivedi

2025-11-04

1 Goal of this Workshop

You'll build a **Custom GPT** that codes qualitative transcripts using your own **codebook**, produces **JSONL-structured outputs** and **Markdown memos**, and aligns findings with the **Relational Coordination (RC)**.

2 Files You Need (Knowledge Pack)

Prepare these before starting:

File	Purpose
`codebook.csv`	List of codes, definitions, inclusion/exclusion, examples
`frameworks.md`	RC framework summary
`style_guide.md`	Formatting rules for memos & outputs
`project_context.md`	Project background & ethical statement
(Optional) transcripts `.txt`	Example interviews or focus groups

3 Step-by-Step: Building Your Custom GPT

3.1 Step 1: Open the Builder

1. Open the ChatGPT Builder in your browser; you need to be a paid user to access it.
2. Go to **ChatGPT → Explore GPTs → Create a GPT.**
3. Choose **Configure** for structured setup.

3.2 Step 2: Name & Describe

- **Name:** `Qualitative Relational Analysis (RC)`
- **Description:** *Codes transcripts with my codebook, outputs JSONL segments + Markdown memos and uses RC.*

3.3 Step 3: Instruction Set (Copy)

Copy and paste the following into the GPT instructions field:

> You are a qualitative analysis assistant for this research project.
>
> **Goals:**
>
> 1. Apply only the uploaded codebook.
> 2. Produce JSONL-formatted coded segments and Markdown memos.
> 3. Cite exact participant quotes with IDs or pseudonyms.
> 4. Map themes from RC using the framework.md.
> 5. Include a short Reflexivity Note in every output.
>
> **Speaker Identification:**
>
> - Read the transcript as written and detect who is speaking (e.g., *Facilitator, A1, A2, Interviewer*).
> - Record that label exactly in `"speaker"`.
> - If no label exists, use `"speaker":"Unknown"`.
> - For overlapping turns, use an array, e.g., `"speaker":["A1","A2"]`.
>
> **Team Identification:**
>
> - When a user provides their coder ID or name (e.g., *KayT, AlexR, C1*), record it in the `"annotator"` field as `"human:<coder_id>"`.
> - If the model generated the code, include its version tag (e.g., `"model:gpt-5-instant"`).
> - Combined example: `"annotator":["model:custom-RCgpt-v1","human:KayT"]`.
>
> **Rules:**
>
> - Use only defined codes; if none fit, assign `"UNSURE"`.
> - Segment length: ~25–120 words; max 3 codes per segment.
> - Include `confidence` (0–1) and a brief `rationale`.
> - Never paraphrase participant text; quote exactly.
>
> **Outputs:**
>
> - **Coding (JSONL, one record per segment):**

```
{
  "doc_id": "INT01",
  "speaker": "Participant",
  "char_span": [123, 256],
  "excerpt": "I only trust it when it matches my gut feeling.",
  "codes": ["Trust.Calibration"],
```

```
  "confidence": 0.82,
  "rationale": "Mentions aligning trust to AI reliability.",
  "annotator": ["model:custom-RCgpt-v1","human:KayT"],
  "timestamp": "2025-11-04T18:15:00Z",
  "framework_tags": ["RC.SharedKnowledge"]
}
```

Summaries (Markdown): include a code-frequency table, 2–3 exemplar quotes per major code, a brief rationale, and a Reflexivity Note.

Frameworks: Use the uploaded `frameworks.md` for RC mapping; justify each link briefly.

Ethics: Anonymize all PII. If content appears sensitive or non-consented, pause and request IRB confirmation.

3.3.1 Why the Example Quote Is There

The short example quote (e.g., *"I only trust it when it matches my gut feeling."*) is **not real data**. It's a *formatting cue* showing the model what a proper `excerpt` string looks like — realistic, quoted, and complete. This ensures the model outputs clean JSON with correct syntax.

3.4 Step 4: Upload Your Knowledge Files

In **Configure → Knowledge**, upload: `codebook.csv`, `frameworks.md`, `style_guide.md`, `project_context.md`, plus any sample transcripts. (Each GPT can store up to ~20 files; keep them concise.)

3.5 Step 5: Set Capabilities

- **Web browsing:** Off (keeps analysis IRB-safe).
- **Code Interpreter:** On (optional; allows quick code counts).
- **File uploads:** On (so you can drop transcripts later).
- **Additional Settings:** Turn off "Use conversatio ndata in your GPT to improve our models"

3.6 Step 6: Save and Test

1. Click **Save** → choose *Only me* (private) or *Anyone with the link* for your workshop.
2. Open your new GPT.
3. Upload a transcript.
4. Prompt:

"Code this transcript using `codebook.csv`; output JSONL with doc_id, speaker, char_span, excerpt, codes, confidence, rationale, framework_tags, annotator, timestamp; then produce a Markdown summary with counts, exemplar quotes, and a Reflexivity Note."

5. Check for correct JSON shape, dynamic speakers, and clean quotes.

4 Troubleshooting

Issue	Likely cause	Fix
`speaker` stays `"Participant"`	Transcript lacks labels	Add prefixes (`A1:`, `Interviewer:`) or preprocess
`annotator` never changes	Update prompt or capture coder ID manually	e.g., `"annotator":["model:gpt-5-instant","human:C1"]`
Missing frameworks	Re-upload `frameworks.md`; shorten definitions	
Invalid JSON (extra commas/brackets)	Model drift	Re-run with JSON example visible in Instructions

5 Multi-User Team Setup

Step	Action
1	Assign each coder an ID (`C1`, `C2`, etc.)
2	Ask each to declare ID once per session ("My coder ID is C1")
3	Model records `"annotator":["model:…","human:C1"]`
4	Merge outputs across coders for reliability tests
5	Keep `coder_map.csv` separately (for privacy)

6 Ethics Tip

Store only pseudonymous coder IDs, never real names or emails, in shared JSONL files.

7 Best Practices for Research Integrity

Practice	Why It Matters
Include timestamps	Reproducibility & version control
Keep Knowledge concise	Better adherence to your codebook

Practice	Why It Matters
Always anonymize transcripts	Protects participant privacy
Include Reflexivity Notes	Satisfies ethical transparency requirements
Version your GPT (v1, v2…)	Maintain analytic traceability

8 Important Links

8.1 Example GPT

- Qualitative Relational Analysis GPT — Example Custom GPT for qualitative analysis

8.2 OpenAI Documentation

- Introducing GPTs — Overview of custom GPTs and how to create them
- Building GPTs Collection — Comprehensive guides on creating and customizing GPTs
- Key Guidelines for Writing Instructions — Best practices for crafting effective GPT instructions
- Custom GPTs Resource — Building repeatable workflows with custom GPTs
- Building Custom GPTs to Automate Tasks — Guide to creating custom GPTs for automation