

From Corpus to Dataset

AI-Human Workflow for Extracting and Validating Data from Large Text and Media Corpora

October 30, 2026 | 3:00–5:00 p.m. | POSCO Hall, Ewha Womans University | English and Korean with AI interpretation

Instructor: Jacob Reidhead (National Chengchi University) | Last revised: July 21, 2026

Workshop overview

This workshop introduces a reproducible, human-supervised workflow for turning large collections of documents, transcripts, articles, webpages, or other media-derived text into structured research datasets. Participants move from exploratory reading to a formal codebook, machine-readable extraction schema, auditable dataset, and preliminary analysis.

The central principle is that an AI model should not silently decide the research design. Researchers define the unit of analysis, variables, evidence rules, and acceptable inferences; AI accelerates document reading, candidate discovery, structured coding, and quality checks.

Intended audience

- Social science and humanities researchers, graduate students, research assistants, librarians, and data journalists working with large text or media corpora.
- Participants may be beginners with AI. Basic familiarity with spreadsheets is sufficient; Python is useful but not required.
- The workshop is especially suitable for projects involving entity, event, relationship, frame, theme, sentiment, or attribute extraction.

Learning outcomes

- Distinguish corpus preparation, retrieval, open-ended discovery, classification, structured extraction, and inference.
- Define a defensible unit of analysis and translate a research question into an explicit variable codebook.
- Design prompts and JSON or tabular schemas that require evidence-linked, machine-readable outputs.
- Run a pilot extraction, identify systematic errors, and revise the codebook before scaling.
- Create a basic validation plan using audit samples, double-coding, agreement measures, and an error log.
- Produce summary statistics and simple visualizations from the extracted dataset.

Workshop outputs

- A corpus inventory with stable document identifiers and source metadata.

- An exploratory typology of candidate entities, events, relationships, themes, or frames.
- A concise codebook defining variables, allowed values, missing-value rules, and evidence requirements.
- A structured extraction prompt and output schema.
- A small pilot dataset with source excerpts and review status.
- A quality-assurance plan and preliminary descriptive analysis.

Before the workshop

Participant preparation

- Bring a laptop and access to at least one capable AI assistant. A paid account is helpful for larger files but is not required.
- Option A: bring a small corpus as CSV/XLSX with one row per document and fields such as document_id, source, date, URL, and document_text.
- Option B: bring a ZIP file containing PDFs, DOCX, TXT, HTML, or transcript files. Remove confidential, restricted, or personally sensitive material unless institutional approval and an appropriate secure environment are in place.
- Option C: use one of the instructor-provided practice corpora.
- Install nothing in advance if using Google Colab. Participants using local Python should have pandas, openpyxl, matplotlib, pydantic, and a PDF-text extraction library available.

Instructor preparation

- Prepare two or three small corpora of approximately 20–100 documents so every participant can complete the workflow within two hours.
- Create a shared folder containing the corpora, a starter notebook, prompt templates, codebook template, example outputs, and troubleshooting notes.
- Test file upload limits, internet access, AI accounts, and the workshop room's Wi-Fi before the session.
- Provide a non-sensitive fallback corpus and a completed example dataset in case a participant's files cannot be parsed.
- Collect participants' topics or data needs in advance through a short registration form when customized corpora will be prepared.

Suggested practice corpora

- Optional comparison corpus — Peter Mandelson: documents from the Epstein files that reference Peter Mandelson, suitable for actor, event, relationship, and evidence extraction.
- Primary workshop corpus — North Korean enterprises: KCNA and Rodong Sinmun articles referencing enterprises, suitable for organization, industry, location, activity, and state-enterprise relationship extraction.
- Participant-defined corpus: a bounded collection prepared in advance from the Epstein or North Korean media databases.
- Public-domain alternative: speeches, parliamentary debates, press releases, news transcripts, or historical documents with stable source metadata.

Core workflow

1. Audit and prepare the corpus

Create stable document IDs, preserve source metadata, extract searchable text, identify languages and duplicates, and record parsing failures.

2. Conduct open-ended discovery

Ask the model to read a bounded sample and identify candidate themes, frames, actors, organizations, events, relations, and recurring ambiguities with supporting excerpts.

3. Align the typology with the research design

Choose the unit of analysis and decide which discovered categories matter theoretically. Define each variable, allowed value, decision rule, and missing or uncertain state.

4. Build and pilot the extraction schema

Require structured JSON or tabular output, evidence excerpts, source identifiers, confidence or uncertainty fields, and no unsupported inference. Pilot on 5–10 diverse documents.

5. Revise, scale, and log

Inspect errors, revise the codebook and prompt, freeze a version, then process the corpus in batches while recording model, prompt, schema, and run identifiers.

6. Validate and analyze

Audit a random and difficult-case sample, compare human and AI coding, adjudicate disagreements, calculate appropriate reliability metrics, and produce descriptive tables and plots.

Two-hour schedule

3:00–3:10 | Welcome, goals, and safeguards

Define the research problem, unit of analysis, expected dataset, and privacy or copyright constraints.

3:10–3:25 | Corpus audit and preparation

Inspect file structure, metadata, text quality, duplicates, language, and missing documents.

3:25–3:45 | Open-ended discovery

Generate candidate entities, events, relationships, themes, frames, and typologies with evidence.

3:45–4:10 | Codebook and schema design

Convert exploratory findings into variables, allowed values, evidence rules, and uncertainty states.

4:10–4:30 | Structured pilot extraction

Run the schema on a small batch, export JSON/CSV, and identify failure patterns.

4:30–4:45 | Quality assurance and reliability

Audit results, double-code a sample, classify errors, and revise the specification.

4:45–4:55 | Preliminary analysis

Calculate frequencies, missingness, cross-tabulations, and simple univariate or bivariate plots.

4:55–5:00 | Wrap-up and next steps

Freeze versions, assign human review, plan scaling, and document limitations.

Detailed itinerary

3:00–3:10 | Frame the extraction task

- Facilitator: contrast search/retrieval, summarization, classification, extraction, and causal or interpretive inference.
- Participants: write one sentence specifying the corpus, unit of analysis, variables of interest, and intended downstream analysis.
- Deliverable: a one-sentence extraction goal and a list of prohibited or unsupported inferences.

3:10–3:25 | Audit the corpus

- Facilitator: demonstrate a corpus inventory and explain why stable IDs and source metadata must survive every transformation.
- Participants: inspect document counts, file types, encodings, languages, OCR quality, duplicates, empty records, and unusually long documents.
- Deliverable: `corpus_inventory.csv` with `document_id`, `source`, `date`, `URL/path`, `language`, `character count`, `parse status`, and `notes`.

3:25–3:45 | Discover candidate categories

- Facilitator: demonstrate an open-ended prompt on a small, diverse sample rather than the entire corpus.
- Participants: request candidate actors, organizations, events, relations, themes, frames, ambiguities, and counterexamples with exact evidence excerpts.

- Deliverable: an exploratory typology and a shortlist of categories that may become variables.

3:45–4:10 | Build the codebook

- Facilitator: show how apparently simple categories conceal unit-of-analysis, threshold, temporality, and inference decisions.
- Participants: define each variable's name, type, definition, allowed values, inclusion/exclusion rules, evidence requirement, uncertainty state, and examples.
- Deliverable: codebook_v1 with no category defined solely by an example.

4:10–4:30 | Run a structured pilot

- Facilitator: demonstrate JSON-schema-constrained output and conversion to a dataframe.
- Participants: extract 5–10 varied documents, including at least one difficult or negative case.
- Deliverable: pilot_extraction.jsonl or pilot_dataset.csv with evidence_quote and review_status fields.

4:30–4:45 | Audit and revise

- Facilitator: introduce false positives, false negatives, boundary errors, unsupported inference, inconsistent normalization, and duplicate entities.
- Participants: manually review a sample, label error types, compare coding, and revise the prompt or codebook rather than patching rows silently.
- Deliverable: error_log.csv and codebook_v2.

4:45–4:55 | Analyze the pilot dataset

- Facilitator: demonstrate missingness, frequency tables, bar charts, histograms, and a simple Y-by-X comparison.
- Participants: inspect whether the extracted variables vary meaningfully and whether missingness reflects the corpus or extraction failure.
- Deliverable: one summary table, one graph, and a list of data-quality concerns.

4:55–5:00 | Freeze and scale

- Facilitator: explain versioning, batch processing, cost tracking, data security, and human-review allocation.
- Participants: record the final model, prompt, schema, codebook, sample, and review plan.
- Deliverable: a reproducible run manifest and next-step checklist.

Reusable prompt templates

Prompt 1: Corpus audit

You are assisting with corpus preparation. Inspect the supplied file inventory and sample text.

Report: (1) document and field structure; (2) missing, empty, duplicate, corrupted, or unusually long

records; (3) language and encoding issues; (4) likely OCR or parsing failures; (5) metadata needed for reproducibility; and (6) a recommended normalized table. Do not analyze substantive content yet. Return a concise audit plus a proposed column schema.

Prompt 2: Open-ended discovery

Read every document in this bounded sample. Identify recurring and exceptional themes, frames, actors, countries, institutions, organizations, events, and relationships. For each candidate category, provide a definition, subtypes, representative evidence excerpts, document IDs, difficult boundary cases, and plausible alternative interpretations. Separate directly observed content from your interpretation. Do not claim corpus-wide prevalence from this sample.

Prompt 3: Codebook critic

Review this proposed codebook as a skeptical methods reviewer. For every variable, identify ambiguity in the unit of analysis, definition, time frame, inclusion/exclusion rule, allowed values, missing-value treatment, evidence requirement, and permitted inference. Propose a revised operational definition and at least one positive, negative, and borderline example. Flag variables that cannot be measured reliably from the available text.

Prompt 4: Structured extraction

Extract only information supported by the document. Follow the supplied codebook and output schema exactly. For each coded value, include `document_id`, `unit_id`, normalized value, exact `evidence_quote`, evidence location when available, and `uncertainty_reason`. Use null or the specified uncertain category when evidence is absent or ambiguous. Do not infer identity, intent, causality, relationships, or attributes beyond the codebook. Return valid JSON only.

Prompt 5: Quality-control reviewer

Compare the document, codebook, and extracted record. Check for false positives, false negatives, incorrect boundaries, unsupported inference, normalization errors, duplicate entities, inconsistent missing values, and evidence that does not support the coded value. Return the proposed correction, `error_type`, severity, rationale, and whether the prompt, codebook, parser, or individual record should be changed.

Recommended data structures

Document table: one row per source document

- `document_id`, source, `publication_date`, author, title, URL or `file_path`, language, `document_text`, `character_count`, `parse_status`, and `corpus_version`.
- Keep the original source file or immutable source reference. Never rely on the cleaned text alone.

Extraction table: one row per unit of analysis

- document_id and unit_id.
- One column per coded variable, using explicit null, not_applicable, uncertain, and not_observed rules.
- evidence_quote, evidence_start, evidence_end, page or timestamp when available.
- model, model_version, prompt_version, schema_version, run_id, extraction_timestamp, and estimated cost.
- confidence should record uncertainty for review prioritization, not substitute for validation.
- review_status, human_coder, adjudication_status, error_type, and notes.

Relationship table: edge-list format

- subject_id, relation_type, object_id, start_date, end_date, document_id, evidence_quote, and review_status.
- Maintain a separate entity table with canonical IDs, aliases, entity types, and disambiguation notes.

Quality assurance and reliability

- Pilot before scaling. Select varied, difficult, negative, and multilingual cases rather than only convenient examples.
- Preserve evidence. Every substantive code should be traceable to an excerpt, page, timestamp, or character span.
- Separate observation from inference. Store directly observed facts separately from researcher-derived classifications.
- Double-code a sample. Compare AI output with at least one human coder, and ideally compare two human coders on a shared subset.
- Use task-appropriate metrics: precision, recall, and F1 for detection; accuracy or class-specific performance for classification; Cohen's kappa or Krippendorff's alpha for categorical agreement; and exact or tolerance-based matching for numeric/date fields.
- Stratify audits by model confidence, document length, language, source, category, and known difficult cases.
- Keep an error taxonomy and revise the upstream specification. Do not silently correct individual rows without logging why the error occurred.
- Freeze versions of the corpus, codebook, prompt, schema, model, and post-processing code before the full run.

Recommended tools

AI assistants and model interfaces

- [ChatGPT](#) — interactive discovery, file-based analysis, coding assistance, and structured extraction prototypes.
- [Claude](#) — long-document analysis, codebook critique, and structured extraction prototypes.

- [Gemini](#) — multimodal and long-context workflows, especially within Google’s ecosystem.
- [NotebookLM](#) — source-grounded exploration and synthesis; best used for discovery and teaching rather than as the sole production extraction pipeline.

Corpus preparation and analysis

- [Google Colab or JupyterLab](#) — reproducible Python notebooks.
- [pandas and openpyxl](#) — CSV/XLSX preparation, cleaning, reshaping, and export.
- [PyMuPDF or pypdf](#) — extracting text and page metadata from PDFs.
- [FFmpeg, Whisper, or institutionally approved transcription tools](#) — converting audio/video into timestamped text when transcripts are unavailable.
- [Pydantic or JSON Schema](#) — validating machine-readable model outputs.
- [matplotlib, plotly, or spreadsheet charts](#) — preliminary descriptive analysis.

Programmatic scaling

- [OpenAI API Structured Outputs or JSON-schema-constrained responses](#).
- [Anthropic API with tool use and structured output validation](#).
- [Gemini API structured output](#).
- [Provider batch APIs for large asynchronous runs when data policy, cost, and deadlines permit](#).
- [Git/GitHub, OSF, or another versioned repository for prompts, codebooks, scripts, manifests, and non-sensitive outputs](#).

Safety, privacy, and research ethics

- Do not upload confidential, personally sensitive, copyrighted, embargoed, or restricted data to a model without authorization and an appropriate data-processing agreement.
- Confirm whether provider settings permit training on submitted data and whether institutional retention, residency, or security requirements are satisfied.
- Minimize data: upload only the fields and excerpts required for the task.
- Treat model output as proposed coding, not ground truth. Human researchers remain responsible for operationalization, interpretation, validation, and publication claims.
- Document exclusions and known blind spots, including OCR quality, language coverage, source selection, missing media, and model changes.
- Avoid using demographic or identity inference unless explicitly justified, ethically approved, and reliably observable from authorized data.

Selected references and documentation

- [Grimmer, Justin, and Brandon M. Stewart. 2013.](#) “Text as Data: The Promise and Pitfalls of Automatic Content Analysis Methods for Political Texts.” *Political Analysis* 21(3): 267–297.
- [Nelson, Laura K. 2020.](#) “Computational Grounded Theory: A Methodological Framework.” *Sociological Methods & Research* 49(1): 3–42.
- [Benoit, Kenneth, et al. 2018.](#) “quanteda: An R Package for the Quantitative Analysis of Textual Data.” *Journal of Open Source Software* 3(30): 774.

- [Roberts, Margaret E., Brandon M. Stewart, Dustin Tingley, et al. 2014.](#) “Structural Topic Models for Open-Ended Survey Responses.” *American Journal of Political Science* 58(4): 1064–1082.
- [Artstein, Ron, and Massimo Poesio. 2008.](#) “Inter-Coder Agreement for Computational Linguistics.” *Computational Linguistics* 34(4): 555–596.
- [Krippendorff, Klaus. Content Analysis: An Introduction to Its Methodology.](#)
- [OpenAI documentation: Structured Outputs and Batch API.](#)
- [Anthropic documentation: Prompt engineering, tool use, and API guides.](#)
- [Google AI for Developers: Structured output documentation for Gemini.](#)
- [Pydantic documentation: model and JSON-schema validation.](#)
- [scikit-learn documentation: classification metrics.](#)

Post-workshop scaling checklist

- Freeze corpus_v1, codebook_v1, prompt_v1, schema_v1, and a gold or adjudicated validation sample.
- Estimate token volume, runtime, cost, concurrency limits, and expected failure rate before submitting the full corpus.
- Choose a batch size and retry policy; log every request and response with stable identifiers.
- Run automatic schema validation and quarantine malformed or incomplete records.
- Review all high-risk categories and a stratified random sample of the remaining records.
- Recompute reliability and performance after every material prompt, codebook, model, or parser change.
- Publish or archive the codebook, prompt, schema, validation procedure, and limitations alongside the dataset when permitted.