Combining MAXQDA, R and Claude

A handbook for frame analysis of global news

Out of Africa project · University of Agder

Version 1.0 · 6 November 2026

RESTRICTED DISTRIBUTION

Companion to the codebook (v1.1), the methodology design paper and the paper "Beyond the Article Text". Menu names and API details reflect November 2026; check them against current documentation.

Restricted distribution · FLL members only · Do not copy or circulate

Contents

How to use this handbook 5

Part I. Foundations 6

1. Purpose and scope . . . 7
   - 1.1 The research problem . . . 7
   - 1.2 What the workflow must deliver . . . 7
   - 1.3 Why three tools . . . 7
2. Theoretical framework in brief . . . 8
   - 2.1 Galtung and Ruge's news factors . . . 8
   - 2.2 Two adaptations for a multipolar news order . . . 8
   - 2.3 Framing . . . 8
   - 2.4 Research questions . . . 9
   - 2.5 Working expectations . . . 9
3. The three tools at a glance . . . 9
   - 3.1 MAXQDA in this project . . . 9
   - 3.2 R in this project . . . 10
   - 3.3 Claude in this project . . . 10
   - 3.4 Decision guide . . . 10

Part II. The integrated workflow 11

4. Overview of the workflow . . . 12
   - 4.1 Folder structure . . . 12
   - 4.2 Identifiers . . . 12
5. Phase 1: Building the corpus . . . 13
   - 5.1 The sampling frame . . . 13
   - 5.2 Running the collection script . . . 13
   - 5.3 Checking what you collected . . . 13
   - 5.4 Preparing files for MAXQDA and R . . . 14
6. Phase 2: Exploratory computational pass in R . . . 14
   - 6.1 Setup . . . 14
   - 6.2 Reading the corpus . . . 14
   - 6.3 Keyword in context . . . 15
   - 6.4 Keyness: what is typical of one outlet . . . 15

RESTRICTED DISTRIBUTION

Restricted distribution · FLL members only · Do not copy or circulate

6.5 Actor dictionary . . . 15
6.6 Attention over time . . . 15
6.7 What to take forward . . . 16

7. Phase 3: Developing the codebook . . . 16
7.1 From theory to code . . . 16
7.2 Building the code system in MAXQDA . . . 16
7.3 Using Claude to stress-test definitions . . . 16
7.4 Anchor examples . . . 17

8. Phase 4: Coding the training sample . . . 17
8.1 Order of work for each article . . . 17
8.2 Article-level codes in MAXQDA . . . 17
8.3 The page track . . . 17
8.4 Memos . . . 18

9. Phase 5: Human reliability . . . 18
9.1 Design . . . 18
9.2 In MAXQDA . . . 18
9.3 In R: Krippendorff's alpha . . . 18
9.4 Reading the result . . . 19

10. Phase 6: Claude as validated third coder . . . 19
10.1 Pilot, validate, admit . . . 19
10.2 Why the API, not the chat . . . 19
10.3 The prompt . . . 19
10.4 Structured output . . . 20
10.5 The coding script . . . 20
10.6 Batches for the full corpus . . . 20
10.7 Screenshots and links . . . 21
10.8 Into MAXQDA . . . 21

11. Phase 7: Integrating and analyzing in R . . . 22
11.1 One data set . . . 22
11.2 B8 link-environment measures . . . 22
11.3 RQ1: Selection . . . 22
11.4 RQ2: Framing of shared events . . . 22
11.5 RQ3: Change and peace journalism . . . 23
11.6 Link networks . . . 23
11.7 Section-front prominence . . . 23

12. Phase 8: Interpreting and writing . . . 24

RESTRICTED DISTRIBUTION

Restricted distribution · FLL members only · Do not copy or circulate

12.1 From pattern to passage . . . 24
12.2 From passage to pattern . . . 24
12.3 Quotations . . . 24
12.4 The methods section . . . 24

Part III. Governance **25**

13. Law, ethics and approvals . . . 26
13.1 Practical rules . . . 26
14. Reproducibility and documentation . . . 27
15. Common pitfalls . . . 27

Appendices **28**

Appendix A. Starter code system . . . 29
Appendix B. Prompt template for Claude . . . 29
Matching JSON schema (excerpt) . . . 29
Appendix C. Checklists by phase . . . 30
Appendix D. Glossary . . . 30
Appendix E. References . . . 31

RESTRICTED DISTRIBUTION

Restricted distribution · FLL members only · Do not copy or circulate

How to use this handbook

This handbook explains how to combine three tools in one research workflow: MAXQDA for qualitative coding, R for computational text analysis and statistics, and Claude for assisted coding, scripting and review. It is written for the *Out of Africa* project. That project compares how six global English-language broadcasters cover East Africa: TRT, CGTN, RT and Press TV, set against CNN International and Al Jazeera English. The method travels well, though. Any frame analysis of news across outlets and over time can use the same setup.

The handbook is practical. Each chapter tells you what to do, in what order, with which tool, and why. Code is included in full. You can copy it, run it and adapt it. Menu names in MAXQDA can shift slightly between versions, so check them against the manual for the version installed at UiA.

There are three parts.

- **Part I. Foundations** sets out the theory, the research questions and what each tool does well and badly.
- **Part II. The integrated workflow** walks through eight phases, from building the corpus to writing up.
- **Part III. Governance** covers copyright, data protection, AI policy, reproducibility and common pitfalls.

The appendices hold a starter code system, complete R and Python scripts, a prompt template for Claude, checklists for each phase, a glossary and references.

> **The governing principle**
>
> People interpret. Machines count. Claude assists both, and is checked by both. Every interpretive claim in the final analysis must rest on a passage a human coder has read in the archived full text. Every number must come from a script that someone else can run again.

Companion files. The handbook assumes these files already delivered to the project: `East_Africa_Corpus_Index.xlsx` (the sampling frame and draft codes), `urls.csv` (the list of addresses), `collect_corpus.py` version 2 (the collection script, with the page track) and `section_fronts.csv`. It also refers to the codebook, version 1.1, and the pilot workbook `Pilot_coding_workbook.xlsx`.

RESTRICTED DISTRIBUTION

Restricted distribution · FLL members only · Do not copy or circulate

Part I. Foundations

RESTRICTED DISTRIBUTION

Restricted distribution · FLL members only · Do not copy or circulate

1. Purpose and scope

1.1 The research problem

Galtung and Ruge argued in 1965 that foreign news follows a set of selection rules. News about the periphery reaches the center only if it is negative, involves elite nations and elite people, and can be told through persons. Sixty years later, the center has split. State-funded broadcasters in Ankara, Beijing, Moscow, Tehran and Doha now compete with CNN and the BBC for the attention of African and global audiences. Scholars call these *contra-flows* (Thussu, 2007).

The project asks whether these contra-flows break the old pattern or only move the center. A first review of coverage from June to November 2026 suggested an answer worth testing. The outlets select much the same East African events. They differ in how they frame them: whom they quote, whom they blame and what they leave out.

1.2 What the workflow must deliver

- A documented, archived corpus of articles with stable identifiers.
- A codebook grounded in theory, with definitions, anchor examples and coding rules.
- Human coding of frames with measured reliability.
- Computational evidence on vocabulary, actors and change over time.
- A transparent, validated use of AI that a reviewer can audit.
- Findings that link numbers to passages and passages to theory.

1.3 Why three tools

No single tool covers all six needs. MAXQDA is built for close reading and coding. R is built for counting, testing and plotting at scale. Claude is fast, reads meaning rather than words, writes code and works in many languages, but it makes confident mistakes and leaves no audit trail of its own. Used together, each covers the others' weaknesses.

RESTRICTED DISTRIBUTION

Restricted distribution · FLL members only · Do not copy or circulate

2. Theoretical framework in brief

2.1 Galtung and Ruge's news factors

Galtung and Ruge (1965) proposed twelve factors that make a foreign event newsworthy. Eight they held to be culture-free. Four they tied to the North-Western news culture.

No.	Factor	Short definition	How it enters the codebook
F1	Frequency	Event time span fits the news cycle	Genre code (flash vs. analysis)
F2	Threshold	Size or intensity of the event	Casualty or scale mentions
F3	Unambiguity	Clear meaning	Complexity code (single vs. multiple causes)
F4	Meaningfulness	Cultural proximity or relevance	Relevance to sponsor state
F5	Consonance	Fits expectations	Fit with sponsor's strategic narrative
F6	Unexpectedness	Surprise within the expected	Novelty marker
F7	Continuity	Stays news once it is news	Diachronic tracking
F8	Composition	Balance of the news mix	Link environment (B8) and section fronts
F9	Elite nations	Reference to powerful states	Sponsor-state reference; foreign actors
F10	Elite people	Reference to powerful persons	Source status (official, expert, citizen)
F11	Personification	Told through persons	Personification code
F12	Negativity	Bad news	Valence code

They also proposed three hypotheses. **Selection and distortion**: each step in the news chain picks out and sharpens these factors. **Additivity**: the more factors, the likelier the event becomes news. **Complementarity**: an event low on one factor must be high on another. Their final section offered remedies: more background, more non-elite actors, more complex and less negative news. Galtung later turned those remedies into the model of peace journalism (Galtung, 1998; Lynch & McGoldrick, 2005).

2.2 Two adaptations for a multipolar news order

1. **From elite nations to sponsor-state reference.** For each outlet, the center is its own sponsor. Code whether and how the sponsor state appears, and note where it is absent though plausibly relevant.
2. **From audience consonance to narrative consonance.** For state outlets, consonance means fit with the sponsor's strategic narrative (Miskimmon, O'Loughlin & Roselle, 2013) more than fit with audience expectations.

2.3 Framing

Entman (1993) defines framing as selecting aspects of a perceived reality and making them salient so as to promote a problem definition, a causal interpretation, a moral evaluation and a treatment recommendation. The codebook turns these four functions into codes. Causal interpretation, or attribution of blame, matters most here.

RESTRICTED DISTRIBUTION

Restricted distribution · FLL members only · Do not copy or circulate

2.4 Research questions

RQ	Focus	Main tool	Support
RQ1	Selection: do the news factors, especially negativity, personification and sponsor-state reference, explain which events each outlet covers?	R (counts, cross-tabs)	MAXQDA codes, Claude coding at scale
RQ2	Framing: how do outlets attribute cause and blame for shared events, and does attribution follow the sponsor's narrative?	MAXQDA (interpretive coding)	R (keyness, collocations), Claude (third coder)
RQ3	Change and norms: how stable are frames over time, and does any outlet approach peace journalism?	R (time series) and MAXQDA (memos)	Claude (summaries per month)

2.5 Working expectations

- **H1.** Events covered by all six outlets score high on several news factors at once.
- **H2.** Outlets differ less in what they select than in whom they name as cause and culprit.
- **H3.** Each outlet references its sponsor state where the reference flatters, and omits it where the state may be implicated.

3. The three tools at a glance

	MAXQDA	R	Claude
Core strength	Close reading and coding of passages	Counting, statistics, plots, reproducibility	Speed, semantic reading, code writing, many languages
Unit of work	Coded segment	Token, document, data frame	Prompted task on a text
Best for	Frames, voices, attribution, memos, reliability	Vocabulary, keyness, actors over time, tests	First-pass coding, third coder, scripting, summaries
Weak at	Statistics, large corpora, rerunnable analysis	Interpretive coding	Consistency, traceability, quotations
Audit trail	Project file with every segment	Scripts and outputs	Only what you log yourself
Learning curve	Low to moderate	Moderate to steep	Low (chat), moderate (Claude Code)
Data location	Local	Local	Anthropic's servers
Role in this project	Place of record for interpretation	Place of record for numbers	Assistant and validated third coder

3.1 MAXQDA in this project

MAXQDA holds the archived texts, the code system, every coded segment and the memos. Document variables (outlet, sponsor, date, topic) let you compare codes across groups. Its team functions support intercoder agreement. Its quantitative module, MAXDictio, offers word frequencies, keyword-in-context and dictionaries, enough for quick checks.

RESTRICTED DISTRIBUTION

Restricted distribution · FLL members only · Do not copy or circulate

3.2 R in this project

R reads the same archived texts and the codes exported from MAXQDA. The `quanteda` family handles tokens, keyword-in-context, keyness and collocations. The `tidyverse` handles data and plots. The `irr` package computes Krippendorff's alpha. Every analysis lives in a script.

3.3 Claude in this project

Claude appears in two forms. The **chat app** suits exploration: testing a code definition on ten articles, discussing a puzzling frame, drafting categories. **Claude Code** works on files and runs scripts in a project folder. It suits production: coding the corpus with a fixed prompt through the API, logging every output, writing and running the R analysis. The API also offers batch processing at half price and structured output that follows a JSON schema, so codes arrive in a fixed format.

3.4 Decision guide

If you need to ...	Use
Decide what a passage means	MAXQDA, by a trained human coder
Count how often a word or actor appears	R (or MAXDictio for a quick look)
Find where to read closely	R keyness and KWIC, or a Claude summary
Draft or stress-test code definitions	Claude chat, then human decision
Apply a settled codebook to hundreds of items	Claude via API, validated against human codes
Measure reliability	MAXQDA (human pairs), R (alpha incl. Claude)
Quote an article in a publication	Archived full text, checked by hand
Produce a figure for a paper	R script under version control
Write or debug code	Claude Code

RESTRICTED DISTRIBUTION

Restricted distribution · FLL members only · Do not copy or circulate

Part II. The integrated workflow

RESTRICTED DISTRIBUTION

Restricted distribution · FLL members only · Do not copy or circulate

4. Overview of the workflow

The workflow has eight phases. Phases move forward, but the codebook loops back: what you learn in coding and validation sharpens the definitions. Freeze a codebook version before each reliability test and before Claude codes at scale.

Phase	Task	Lead tool	Output
1	Build and archive the corpus	Python script	Text, screenshots, links table, logs
2	Explore computationally	R	KWIC lists, keyness tables, attention plots
3	Develop the codebook	MAXQDA + Claude chat	Codebook v1.x with anchor examples
4	Code the training sample	MAXQDA	Coded segments, memos
5	Test human reliability	MAXQDA + R	Agreement report, revised codebook
6	Validate and deploy Claude as third coder	Claude Code (API)	Validated machine codes, logs
7	Integrate and analyze	R	Cross-tabs, models, time series, networks, figures
8	Interpret and write	MAXQDA + R + human judgment	Findings, verified quotations, methods text

4.1 Folder structure

Keep one project folder. Put it under version control with git, or at least in a synced, backed-up location. Never edit the raw texts after archiving.

```
out-of-africa/
  00_admin/           data management plan, approvals, codebook versions
  01_corpus/
     urls.csv         the sampling frame (input)
     section_fronts.csv
     corpus/          output of collect_corpus.py (do not edit)
        html/  text/  json/  page/  links/  fronts/
        retrieval_log.csv  page_log.csv  links_table.csv  fronts_log.csv
  02_maxqda/          MAXQDA project files (.mx24 etc.) and exports
  03_r/               R scripts (numbered), R project file
  04_claude/
     prompts/         versioned prompt files
     outputs/         raw JSON responses, one file per run
     logs/            run logs with model, prompt version, date
  05_results/         tables and figures for publication
  CLAUDE.md           project notes read by Claude Code
  README.md           how to reproduce everything
```

4.2 Identifiers

Every article keeps one ID from start to finish: `T01` for TRT, `C01` for CGTN, `R01` for RT, `P01` for Press TV, `N01` for CNN and `A01` for Al Jazeera. The collection script names files by ID. Use the same ID as the document name in MAXQDA and as the key in R. This single rule makes the three tools join cleanly.

RESTRICTED DISTRIBUTION

Restricted distribution · FLL members only · Do not copy or circulate

5. Phase 1: Building the corpus

5.1 The sampling frame

Start from `East_Africa_Corpus_Index.xlsx`. It lists 85 items found by search across the six outlets for June to early November 2026. Treat it as a first sampling frame, not a census. Before final data collection, decide on a systematic sampling rule. Two workable options:

- **Constructed weeks.** Draw one Monday, one Tuesday and so on from different weeks across the period, and collect all East Africa items from each outlet's Africa section on those days. This controls for weekly news cycles.
- **Event-anchored windows.** Choose the shared events (Sudan, Tigray and Eritrea, Ebola, Somaliland) and collect all items from each outlet within a fixed window around key dates. This suits the same-event comparison best.

Write the rule down before collecting. Record which section pages and search strategies you used for each outlet.

5.2 Running the collection script

The script has two tracks. The **text track** saves the raw page, the extracted article text and a JSON record. The **page track** opens each article in a headless browser and saves a desktop screenshot (1,440 px wide), a mobile screenshot (390 px wide), the page as rendered, and every link around the article, sorted into zones such as "related", "most read" and in-text links. The page track feeds codebook blocks B7 and B8.

```
pip install trafilatura requests playwright
python -m playwright install chromium

python collect_corpus.py articles urls.csv --page --delay 3
python collect_corpus.py articles urls.csv --page --wayback      # second pass for failures
python collect_corpus.py fronts section_fronts.csv               # once a day, same hour
```

The script follows redirects, checks robots.txt, rejects non-essential cookies where it can and logs every step. Text attempts go to `retrieval_log.csv`, page captures to `page_log.csv`, links to `links_table.csv`. Re-running skips what is already archived.

> **Check the zone rules first**
>
> Run `--page --only T01,C01,R01,P01,N01,A01 --out test` and compare each links file with its screenshot. Outlet layouts change. Fix the selectors where a zone is missed, raise the rules version, and record it.

5.3 Checking what you collected

1. Open the logs. Sort by status. List every failure.
2. Open ten text files at random. Check that the extraction caught the full article and nothing else: no menus, no related-story lists.
3. Open the same ten screenshots. Check that the cookie banner is gone and the page loaded to the bottom.
4. Compare the extracted date with the date in the index. Correct the index where they differ, and note which you trust.

Restricted distribution · FLL members only · Do not copy or circulate

5 For failures, try the Internet Archive option, then manual saving in a browser (save as PDF and as HTML), noting the method.

6 Freeze the corpus. Record the number of items per outlet and the date of freezing.

> **Known access issues**
> In November 2026, rt.com redirected to a mirror domain, and RT is under EU sanctions, so access from Norway may vary. Press TV publishes on presstv.ir and presstv.co.uk, and some pages disallow automated access. Do not route around access restrictions. Log the failure, try the Internet Archive, and treat remaining gaps as a limitation you report.

5.4 Preparing files for MAXQDA and R

The `text/` folder holds plain-text files named by ID with a short metadata header. These import directly into MAXQDA. The `json/` folder holds the same text with metadata as fields. R reads these most easily. Screenshots in `page/` can be imported into MAXQDA as image documents named `T01_desktop` and so on.

6. Phase 2: Exploratory computational pass in R

Before anyone codes, let R show you where to look. This phase is fast and cheap. It points close reading to the passages and outlets where vocabulary differs most. It is not evidence yet.

6.1 Setup

```
install.packages(c("tidyverse", "jsonlite", "quanteda", "quanteda.textstats",
                   "quanteda.textplots", "readxl", "irr", "lubridate", "igraph"))
```

6.2 Reading the corpus

```
library(tidyverse); library(jsonlite); library(quanteda)

files <- list.files("01_corpus/corpus/json", pattern = "\\.json$", full.names = TRUE)
docs  <- map_dfr(files, function(f) {
  j <- fromJSON(f)
  tibble(id = j$id, outlet = j$outlet, date = j$date %||% NA_character_,
         title = j$title %||% NA_character_, text = j$text)
})

# Join metadata from the corpus index (topic, sponsor, outlet group)
idx  <- readxl::read_excel("01_corpus/East_Africa_Corpus_Index.xlsx", sheet = "Corpus") |>
        select(id = ID, sponsor = `Sponsor state`, group = `Outlet group`, topic = Topic)
docs <- left_join(docs, idx, by = "id") |>
        mutate(date = lubridate::as_date(date), month = lubridate::floor_date(date, "month"))

corp <- corpus(docs, docid_field = "id", text_field = "text")
toks <- tokens(corp, remove_punct = TRUE, remove_numbers = TRUE) |>
        tokens_tolower() |>
        tokens_remove(stopwords("en"))
```

RESTRICTED DISTRIBUTION

Restricted distribution · FLL members only · Do not copy or circulate

6.3 Keyword in context

Keyword-in-context lists show every use of a term with its surroundings. Use them to see how outlets label actors.

```
k <- kwic(tokens(corp), pattern = c("regime", "militia*", "rebel*", "terroris*"), window = 8)
k_df <- as_tibble(k) |> left_join(select(docs, id, outlet), by = c("docname" = "id"))
count(k_df, outlet, keyword, sort = TRUE)
write_csv(k_df, "05_results/kwic_labels.csv")
```

6.4 Keyness: what is typical of one outlet

Keyness compares word frequencies in a target group against all others. A high score means the word is unusually frequent in the target.

```
library(quanteda.textstats); library(quanteda.textplots)
dfmat <- dfm(toks)
key_ptv <- textstat_keyness(dfmat, target = docvars(dfmat, "outlet") == "Press TV")
textplot_keyness(key_ptv, n = 20)
ggsave("05_results/keyness_presstv.png", width = 8, height = 6, dpi = 300)
```

6.5 Actor dictionary

A small dictionary counts references to foreign actors. This gives a first view of sponsor-state reference and of which outside powers each outlet names.

```
actors <- dictionary(list(
  turkey  = c("turkey", "türkiye", "turkish", "ankara", "erdogan"),
  china   = c("china", "chinese", "beijing", "xi"),
  russia  = c("russia", "russian", "moscow", "kremlin", "putin"),
  iran    = c("iran", "iranian", "tehran"),
  usa     = c("united states", "us", "american", "washington", "trump"),
  uae     = c("uae", "emirati", "emirates", "abu dhabi"),
  israel  = c("israel", "israeli", "tel aviv", "netanyahu"),
  egypt   = c("egypt", "egyptian", "cairo"),
  qatar   = c("qatar", "qatari", "doha")))
act <- tokens(corp) |> tokens_tolower() |> tokens_lookup(actors) |> dfm() |>
       convert(to = "data.frame") |> as_tibble() |>
       left_join(select(docs, id, outlet), by = c("doc_id" = "id"))
act |> group_by(outlet) |> summarise(across(turkey:qatar, ~ mean(.x > 0))) |>
       write_csv("05_results/actor_presence_share.csv")
```

> **Ambiguous terms**
>
> Short forms such as `us` collide with ordinary words. Check matches with KWIC before trusting the counts, and refine the dictionary. Record each change in the script's comments.

6.6 Attention over time

```
docs |> count(outlet, month, topic) |>
  ggplot(aes(month, n, fill = topic)) + geom_col() + facet_wrap(~ outlet, ncol = 2) +
  labs(x = NULL, y = "Articles", fill = "Topic") + theme_minimal()
ggsave("05_results/attention_by_month.png", width = 9, height = 7, dpi = 300)
```

RESTRICTED DISTRIBUTION

Restricted distribution · FLL members only · Do not copy or circulate

6.7 What to take forward

- A list of labels and actor terms that differ most by outlet.
- Ten to twenty articles flagged for close reading because they are typical or extreme.
- Candidate codes that the theory did not anticipate.

7. Phase 3: Developing the codebook

The codebook is the project's central instrument. It turns theory into decisions a coder can make the same way twice. Version 1.1 exists as a shared document. It has five layers: metadata (Part A), quantitative content analysis (Part B, including B7 visual and layout and B8 link environment), three strands of critical discourse analysis (C1 Foucault, C2 Fairclough and Wodak, C3 van Dijk) and the frame synthesis (Part F).

7.1 From theory to code

Every code needs five things. Without all five, two coders will not agree.

Element	What it says	Example (B5.2 Causal attribution)
Name and number	A stable label	B5.2 Causal attribution
Definition	What the code captures, in one sentence	Who or what the text holds responsible for the problem
Values	A closed list, with 99 for cannot determine	1 local government · 2 local armed group · … · 10 none
Coding rule	How to decide hard cases	Code the dominant attribution; headline and lead weigh more
Anchor example	A real passage that clearly fits	Added from the training sample, with article ID

7.2 Building the code system in MAXQDA

1. Create a new project and import the archived text files. Each document name must be the article ID.
2. Create document groups by outlet, so the six outlets can be compared.
3. Import Part A as document variables (Variables, then Import Document Variables) from the corpus index. The spreadsheet needs a column for the document group and one for the document name.
4. Build the code tree in the order of the codebook's Appendix A. Article-level codes get value subcodes. Segment codes (C2) are applied to the shortest span that carries the feature.
5. Attach the definition, rule and anchor to each code as a code memo. Coders then see them on hover.

7.3 Using Claude to stress-test definitions

Claude is useful at this stage as a critic, not as a coder. Give it a definition and a handful of articles. Ask where the definition fails. A typical prompt:

```
Here is a code definition from our codebook, followed by six news articles.
Code each article with it. Then tell me:
1. Which articles were hard to code, and why.
2. Which values were missing.
3. How the definition could be reworded so two coders would agree.
Do not invent quotations; refer to paragraphs by number.
```

RESTRICTED DISTRIBUTION

Restricted distribution · FLL members only · Do not copy or circulate

The pilot of 6 November 2026 did exactly this on 13 articles. It found real gaps. The remedy variable lacked a value for public-health measures. The peace and war indices made no sense for epidemic stories. Ebola coverage fitted none of the starting frames. The full list is in the pilot workbook, sheet Pilot_findings. Each finding is a candidate change for version 1.2. A human decides each one.

> **Who decides**
> Claude proposes. The project team decides. Record every change in the codebook's version log with its date, reason and the codes affected. Never change a definition silently in the middle of coding.

7.4 Anchor examples

Anchor examples come from the corpus, not from Claude. While coding the training sample, mark one clear case per value. Copy the article ID and paragraph number into the code memo. Quote no more than the coder needs to recognize the case.

8. Phase 4: Coding the training sample

The training sample teaches coders the codebook and shows where it is still unclear. Use about 60 articles, ten per outlet, drawn across the period and the event clusters. Include the same-event sets.

8.1 Order of work for each article

1. Read the whole article once without coding.
2. Check Part A against the archived file and correct the document variables.
3. Code Part B at article level, then actors and sources one by one.
4. Open the desktop and mobile screenshots. Code B7.
5. Check that the article's rows in the links table are coded on L0 to L8.
6. For subsample articles: code C2 segments, extract C3 propositions, then code C1 and write the discourse memo.
7. Assign the frame (Part F) and write the two-sentence frame memo.

8.2 Article-level codes in MAXQDA

Article-level codes are applied to the headline segment. This keeps them in one place and makes them easy to export. Alternatively, keep article-level codes as document variables. That is simpler for the quantitative layers and is the route used for Claude's codes (chapter 10). Choose one route before coding starts and keep to it.

8.3 The page track

Screenshots and links add two layers MAXQDA was not built around. Handle them like this.

- **Screenshots.** Import them as image documents in a separate document group per outlet. Code B7 as document variables on the text document. Use image coding only for the qualitative reading of captions and images.
- **Links table.** Code L0 to L8 in the spreadsheet `links_table.csv`, not in MAXQDA. R computes B8.1 to B8.8 from it and writes them back as document variables.

RESTRICTED DISTRIBUTION

Restricted distribution · FLL members only · Do not copy or circulate

- **Section fronts.** These are not coded in MAXQDA. R reads the daily link files and measures prominence: rank, position on the page and whether the link has an image.

8.4 Memos

Three memo types carry the interpretive work. A **discourse memo** (C1) says what the article treats as true, who may say it, and what must be assumed. A **frame memo** (F) names the frame and cites the strongest evidence. A **codebook memo** proposes a change to a definition. Review codebook memos weekly.

9. Phase 5: Human reliability

Reliability shows that the codebook, not the coder, produces the codes. Test it before any machine coding, and again after each major revision.

9.1 Design

- Two coders code the same 30 or more articles independently, stratified by outlet.
- Freeze the codebook version before the test. Record it.
- Coders do not discuss the articles until both have finished.

9.2 In MAXQDA

Each coder works in a copy of the same project with identical documents. Afterwards, open both and run the Intercoder Agreement function from the Analysis menu. Choose the level: code presence per document, code frequency, or segment overlap (60 to 90 percent minimum overlap for C2 segment codes). MAXQDA reports percent agreement and kappa variants. Export the table.

9.3 In R: Krippendorff's alpha

For article-level variables, alpha is the standard measure in content analysis (Hayes & Krippendorff, 2007). Export each coder's codes as one row per article and compute alpha per variable.

```
library(tidyverse); library(irr)
c1 <- read_csv("02_maxqda/exports/coder1_article_codes.csv")
c2 <- read_csv("02_maxqda/exports/coder2_article_codes.csv")
vars <- setdiff(intersect(names(c1), names(c2)), "id")

alpha_for <- function(v, level = "nominal") {
  m <- rbind(c1[[v]], c2[[v]])
  m[m == 99] <- NA                              # 'cannot determine' as missing
  kripp.alpha(m, method = level)$value
}
ordinal <- c("B1_8", "B6_peace", "B6_war", "F3_3")
rel <- tibble(variable = vars,
              level = if_else(vars %in% ordinal, "ordinal", "nominal")) |>
       mutate(alpha = map2_dbl(variable, level, alpha_for))
write_csv(rel, "05_results/reliability_human.csv")
```

RESTRICTED DISTRIBUTION

Restricted distribution · FLL members only · Do not copy or circulate

9.4 Reading the result

Alpha	Decision
0.80 or higher	Keep the variable as defined
0.67 to 0.80	Keep for tentative conclusions; sharpen the rule and retest
Below 0.67	Revise the definition or drop the variable

Discuss every disagreement. Log what caused it: an unclear rule, a missing value, or a hard text. Revise the codebook, raise the version number and retest the weak variables on fresh articles.

10. Phase 6: Claude as validated third coder

Once human reliability is good, Claude can code the article-level variables of Parts A, B and B7 across the full corpus. It does so only for variables it has been shown to code as well as trained humans. The interpretive layers C1, C2, C3 and F stay with human coders. Claude may suggest; a human decides.

10.1 Pilot, validate, admit

Step	What happens	Rule
Pilot	Claude codes 10 to 15 articles in chat; the team inspects	Find gaps, not measure accuracy
Freeze	Codebook version, prompt version, model and output schema fixed	Logged in 04_claude/logs
Validate	Two humans code 60+ articles; consensus is the reference. Claude codes the same set twice, on different days	Alpha vs. consensus; stability between runs
Admit	A variable is machine-coded only if alpha is .80 or higher and the two runs agree on 90 percent of items	Per variable, not per codebook
Audit	Humans check all low-confidence items and a random 10 percent	Audit alpha below .80 stops machine coding of that variable

The pilot is done. Claude coded 13 articles; the workbook holds a blind human-coding sheet and an agreement sheet for the comparison. Treat its codes as material for discussion, not data.

10.2 Why the API, not the chat

A chat session carries memory from one article to the next. It cannot be rerun in the same way and leaves no clean log. The production run uses the API, called by a script: one article per call, the same frozen prompt every time, and every input and output saved. Claude Code can write and run that script in the project folder.

10.3 The prompt

Keep the prompt in `04_claude/prompts/coding_v1.1.md`. It holds the role, the general rules, the definitions and values of every variable to be coded, and an instruction to answer only in the given format. Appendix B gives a template.

RESTRICTED DISTRIBUTION

Restricted distribution · FLL members only · Do not copy or circulate

10.4 Structured output

The API can be told to answer in a fixed JSON schema. Every reply then has the same fields, with values from the allowed list. This removes most parsing errors. Each variable gets its value, a confidence rating and, for subtle variables, a one-line reason that cites a paragraph number.

10.5 The coding script

The script below reads the archived text files, sends each to Claude with the frozen prompt and schema, and saves the raw reply and a log line. Set the API key in the environment variable ANTHROPIC_API_KEY. Do not write it in the script.

```
# code_corpus.py  -  Claude as third coder (article-level variables)
import json, csv, pathlib, datetime, anthropic

MODEL    = "claude-sonnet-5-5"           # pin the model; record it in the log
PROMPT_V = "coding_v1.1"
PROMPT   = pathlib.Path(f"04_claude/prompts/{PROMPT_V}.md").read_text(encoding="utf-8")
SCHEMA   = json.loads(pathlib.Path("04_claude/prompts/schema_v1.1.json").read_text())
OUT      = pathlib.Path("04_claude/outputs") / PROMPT_V
OUT.mkdir(parents=True, exist_ok=True)
client   = anthropic.Anthropic()

def code_article(path):
    text = path.read_text(encoding="utf-8")
    msg = client.messages.create(
        model=MODEL, max_tokens=2000, temperature=0,
        system=PROMPT,
        messages=[{"role": "user", "content": f"<article id='{path.stem}'>\n{text}\n</article>"}],
        output_config={"format": {"type": "json_schema", "schema": SCHEMA}},
    )
    reply = next(b.text for b in msg.content if b.type == "text")
    return json.loads(reply), msg

with open("04_claude/logs/runs.csv", "a", newline="") as logf:
    log = csv.writer(logf)
    for f in sorted(pathlib.Path("01_corpus/corpus/text").glob("*.txt")):
        target = OUT / f"{f.stem}.json"
        if target.exists():
            continue
        codes, msg = code_article(f)
        target.write_text(json.dumps(codes, indent=2, ensure_ascii=False), encoding="utf-8")
        log.writerow([f.stem, MODEL, PROMPT_V, "1.1",
                      datetime.datetime.now(datetime.timezone.utc).isoformat(),
                      msg.usage.input_tokens, msg.usage.output_tokens])
        print(f.stem, "done")
```

10.6 Batches for the full corpus

For hundreds of articles, the Message Batches API costs half the standard price. Most batches finish within an hour; a batch expires if it is not done in 24 hours, and results can be downloaded for 29 days. Use the article ID as the custom ID, because results do not come back in order.

RESTRICTED DISTRIBUTION

Restricted distribution · FLL members only · Do not copy or circulate

```
from anthropic.types.message_create_params import MessageCreateParamsNonStreaming
from anthropic.types.messages.batch_create_params import Request

reqs = [Request(custom_id=f.stem,
                params=MessageCreateParamsNonStreaming(
                    model=MODEL, max_tokens=2000, temperature=0, system=PROMPT,
                    messages=[{"role": "user",
                               "content": f"<article id='{f.stem}'>\n{f.read_text()}\n</article>"}],
                    output_config={"format": {"type": "json_schema", "schema": SCHEMA}}))
        for f in sorted(pathlib.Path("01_corpus/corpus/text").glob("*.txt"))]
batch = client.messages.batches.create(requests=reqs)
print("batch id:", batch.id)        # save it in the log

# later:
for r in client.messages.batches.results(batch.id):
    if r.result.type == "succeeded":
        reply = next(b.text for b in r.result.message.content if b.type == "text")
        (OUT / f"{r.custom_id}.json").write_text(reply, encoding="utf-8")
    else:
        print(r.custom_id, r.result.type)      # errored or expired: resend
```

10.7 Screenshots and links

B7 variables are coded from the screenshots. The API accepts images, so the same script can send the desktop and mobile captures with the text. Link-level variables L1 to L8 are coded one link per call from the headline, then aggregated in R. Both need their own validation round: an image prompt is a different instrument from a text prompt.

10.8 Into MAXQDA

Assemble the JSON outputs into one spreadsheet, one row per article, with the document group and document name in the first two columns. Add a column naming the variables that were machine-coded, and the prompt version. Import it as document variables. The pilot file `Pilot_MAXQDA_document_variables.xlsx` shows the layout.

```
import json, pathlib, pandas as pd
rows = []
for f in sorted(pathlib.Path("04_claude/outputs/coding_v1.1").glob("*.json")):
    d = json.loads(f.read_text())
    flat = {k: v["value"] for k, v in d["codes"].items()}
    flat.update({"Document name": f.stem, "machine_coded": 1, "prompt_version": "coding_v1.1"})
    rows.append(flat)
df = pd.DataFrame(rows)
idx = pd.read_excel("01_corpus/East_Africa_Corpus_Index.xlsx", sheet_name="Corpus")[["ID",
    "Outlet"]]
df = df.merge(idx, left_on="Document name", right_on="ID").rename(columns={"Outlet": "Document
    group"})
cols = ["Document group", "Document name"] + [c for c in df.columns if c not in ("Document group",
    "Document name", "ID")]
df[cols].to_excel("02_maxqda/import/claude_document_variables.xlsx", index=False)
```

RESTRICTED DISTRIBUTION

Restricted distribution · FLL members only · Do not copy or circulate

11. Phase 7: Integrating and analyzing in R

R joins the three sources of codes on the article ID: human codes exported from MAXQDA, validated Claude codes, and the measures computed from the page track. Then it answers the research questions.

11.1 One data set

```
library(tidyverse); library(readxl)
human  <- read_excel("02_maxqda/exports/document_variables.xlsx") |> rename(id = `Document name`)
claude <- read_excel("02_maxqda/import/claude_document_variables.xlsx") |> rename(id = `Document
    name`)
links  <- read_csv("01_corpus/corpus/links_table.csv")

# Human codes win. Claude fills only admitted variables that humans did not code.
admitted <- read_csv("05_results/admitted_variables.csv")$variable
fill     <- setdiff(admitted, names(human))
art <- human |> left_join(select(claude, id, any_of(fill)), by = "id")
```

11.2 B8 link-environment measures

```
rel <- links |>
  filter(capture == "desktop", zone %in% c("related_sidebar", "related_end", "read_more_box",
    "in_text"),
         curator != "popularity", L7_link_type != 6)
b8 <- rel |> group_by(parent_id) |>
  summarise(B8_7 = n(),
            B8_1 = mean(L2_region %in% c(1, 2)),
            B8_2 = mean(L4_sponsor_ref == 1 | L7_link_type == 4),
            B8_3 = mean(L7_link_type == 3 & !(L2_region %in% c(1, 2, 3))),
            B8_4 = sum(L7_link_type %in% c(1, 2)) / max(1, sum(L7_link_type %in% 1:3)),
            B8_6 = mean(L8_crisis == 1)) |>
  mutate(across(B8_1:B8_6, ~ if_else(B8_7 < 3, NA_real_, .x)))   # too few links: missing
art <- left_join(art, b8, by = c("id" = "parent_id"))
```

11.3 RQ1: Selection

Cross-tabulate news factors by outlet. Test differences with chi-square tests, or with logistic regression when several factors are tested together. Report effect sizes, not only p-values.

```
tab <- table(art$A2_outlet, art$B1_5)          # sponsor-state reference by outlet
chisq.test(tab); prop.table(tab, 1)

m1 <- glm(I(B1_8 == 1) ~ A4_group + A7_cluster + A10_genre, data = art, family = binomial)
summary(m1)
```

11.4 RQ2: Framing of shared events

Restrict the data to same-event sets (variable A13). Compare causal attribution, sources and frames within each set. This holds the event constant, so differences are differences in framing.

```
same <- art |> filter(!is.na(A13_event), A13_event != "")
same |> count(A13_event, A2_outlet, B5_2) |> pivot_wider(names_from = B5_2, values_from = n,
    values_fill = 0)
same |> count(A2_outlet, F3_1) |> group_by(A2_outlet) |> mutate(share = n / sum(n))
```

RESTRICTED DISTRIBUTION

Restricted distribution · FLL members only · Do not copy or circulate

11.5 RQ3: Change and peace journalism

```
art |> mutate(month = lubridate::floor_date(as.Date(A5_date), "month")) |>
  filter(B2_1 %in% c(1, 2, 7, 8)) |>                # conflict items only (B6 rule)
  group_by(A2_outlet, month) |> summarise(peace = mean(B6_peace), war = mean(B6_war), n = n()) |>
  pivot_longer(c(peace, war)) |>
  ggplot(aes(month, value, color = name)) + geom_line() + facet_wrap(~ A2_outlet) +
  labs(x = NULL, y = "Mean index (0–5)", color = NULL) + theme_minimal()
ggsave("05_results/peace_war_by_month.png", width = 9, height = 6, dpi = 300)
```

11.6 Link networks

Each outlet's links form a network of stories. Communities show which events an outlet habitually places together.

```
library(igraph)
net_for <- function(o) {
  e <- rel |> filter(outlet == o, internal == 1) |> select(parent_id, target_url)
  g <- graph_from_data_frame(e, directed = TRUE)
  list(graph = g, communities = cluster_walktrap(as.undirected(g)),
       hubs = sort(degree(g, mode = "in"), decreasing = TRUE)[1:10])
}
nets <- set_names(unique(rel$outlet)) |> map(net_for)
```

11.7 Section-front prominence

```
fronts <- list.files("01_corpus/corpus/fronts", pattern = "_desktop_links\\.csv$",
                     recursive = TRUE, full.names = TRUE) |> map_dfr(read_csv)
urls <- map_dfr(files, ~ { j <- fromJSON(.x); tibble(id = j$id, target_url = j$url_retrieved) })
fronts |> left_join(urls, by = "target_url") |>
  group_by(outlet, date) |>
  summarise(corpus_items_on_front = sum(!is.na(id)),
            best_rank = suppressWarnings(min(rank[!is.na(id)])), .groups = "drop")
```

RESTRICTED DISTRIBUTION

Restricted distribution · FLL members only · Do not copy or circulate

12. Phase 8: Interpreting and writing

This is where the numbers and the readings meet. The rule is simple. Every pattern in the counts is explained by passages. Every reading of a passage is checked against the counts.

12.1 From pattern to passage

1. Take a pattern from R, for example that one outlet attributes blame to Western states far more often than others.
2. List the articles behind it. Open them in MAXQDA.
3. Read the coded segments for B5.2 and the C2 strategies. Find what produces the pattern in the text.
4. Write the finding as a claim, its number, and one or two anchored passages.

12.2 From passage to pattern

A close reading may suggest a pattern, such as a habitual pairing of an East African crisis with a sponsor's aid story. Test it in the links table or the codes across the whole corpus before claiming it.

12.3 Quotations

- Copy every quotation from the archived file, never from Claude's output or a memo.
- Check it against the live page or the Internet Archive snapshot where possible.
- Cite article ID, outlet and date. Keep quotations short.

12.4 The methods section

Report: the sampling rule and its result; the capture dates and the access failures; the codebook version; human reliability per variable; which variables Claude coded, its validation alpha and audit results; the model and prompt version; and the limits of the page track (capture time, location, personalization).

> **A sentence for the methods section**
>
> Article-level variables in Parts A and B were coded by Claude (model and prompt version stated) after validation against the consensus of two trained human coders on a stratified set of 60 articles (Krippendorff's alpha ≥ .80 for each admitted variable). All interpretive codes were assigned by human coders.

RESTRICTED DISTRIBUTION

Restricted distribution · FLL members only · Do not copy or circulate

Part III. Governance

RESTRICTED DISTRIBUTION

Restricted distribution · FLL members only · Do not copy or circulate

13. Law, ethics and approvals

Clear four questions with UiA before collection begins. Record each answer in the data management plan.

Issue	The question	Where to ask
Copyright	May the project copy and store full articles and screenshots for analysis? On what terms may excerpts be published?	UiA library or legal office
Personal data	Articles and photographs name and show people. Does the project need a notification to Sikt, and what storage rules apply?	Sikt; UiA data protection officer
AI services	May archived texts and screenshots be sent to an external AI service? Which agreement and settings are required?	UiA IT and research administration
Sanctions	RT is subject to EU sanctions on broadcasting. Does accessing and storing RT pages for research raise any issue in Norway?	UiA legal office

13.1 Practical rules

- Store the corpus, screenshots and Claude outputs in the access-controlled project area. Never in public repositories.
- Share the links table and codes as research data; share texts and screenshots only as the approvals allow.
- Do not publish images of private individuals. Code categories, not identities.
- Respect robots.txt and site terms. Do not log in or circumvent access restrictions.
- Space requests in time so the study places negligible load on the outlets' servers.

RESTRICTED DISTRIBUTION

Restricted distribution · FLL members only · Do not copy or circulate

14. Reproducibility and documentation

- **Versions.** Codebook, prompt, collection rules and model are versioned and named in every output.
- **Scripts.** Every number comes from a numbered script in 03_r or 04_claude. No manual edits to outputs.
- **Logs.** Retrieval, page and front logs; Claude run logs with model, prompt version, date and token counts.
- **README.** One page that tells a newcomer how to rerun the whole chain.
- **CLAUDE.md.** Project notes that Claude Code reads first: folder layout, ID rule, what never to edit.

```
# CLAUDE.md (excerpt)
- Article IDs (T01, C01 ...) are the key everywhere. Never rename files.
- Never edit anything in 01_corpus/corpus/. It is the frozen archive.
- Coding prompts live in 04_claude/prompts/. Never change a frozen version; copy and raise the
    number.
- Every Claude coding run appends to 04_claude/logs/runs.csv.
- R scripts are numbered by phase. Outputs go to 05_results/.
```

15. Common pitfalls

Pitfall	Remedy
Coding from Claude's summary instead of the text	Code only from the archived file
Changing a definition during coding without a version	Codebook memo, weekly review, new version
Treating percent agreement as reliability	Report Krippendorff's alpha per variable
Admitting machine coding for the whole codebook at once	Admit variable by variable
Reading algorithmic link zones as editorial framing	Use the curator code; interpret by zone
Comparing link environments captured weeks apart	Report capture lag (A14); capture promptly
Mobile layout missing from the analysis	Code B7.11; report what mobile readers see
Quoting from memory or from an AI output	Copy from the archive; verify against the source
A dictionary term that matches ordinary words	Check with KWIC before counting
Wire copy treated as the outlet's own reporting	Code A9; compare what each outlet changed

RESTRICTED DISTRIBUTION

Restricted distribution · FLL members only · Do not copy or circulate

Appendices

RESTRICTED DISTRIBUTION

Restricted distribution · FLL members only · Do not copy or circulate

Appendix A. Starter code system

The full code system, with definitions and values, is in the codebook, version 1.1, Appendix A. The outline below shows the structure to build in MAXQDA.

```
A  Metadata (document variables)      A1–A16
B  QCA (article level)
   B1 News factors        B1.1–B1.10
   B2 Issue               main, secondary
   B3 Actors              role, evaluation, label
   B4 Sources             type, origin, mode, gender; B4.5–B4.7
   B5 Framing functions   problem, cause, attribution mode, evaluation, remedy
   B6 Peace/war journalism B6.1–B6.10 (conflict items only)
   B7 Visual and layout   B7.1–B7.11 (screenshots)
   B8 Link environment    B8.1–B8.7 (computed in R from L0–L8)
C1 Foucault               truth regime, exclusion, subject positions, normalization, legitimation
C2 Fairclough and Wodak   vocabulary, grammar, DHA strategies, topoi, intertextuality
C3 van Dijk               propositions, macrostructure, local semantics, ideological square
F  Frames                 dominant, secondary, strength, contest
Memos                     discourse, explanation, frame, codebook
```

Appendix B. Prompt template for Claude

```
You are a trained content-analysis coder in the Out of Africa project.
You code one news article at a time, using codebook version 1.1.

General rules
- Code what the text does, not what the outlet is. Ownership is never evidence.
- Code only what is explicitly present (manifest). Do not infer.
- Headline and lead weigh more than the body when a variable asks for the dominant value.
- Use 99 when the text does not let you decide. Never guess.
- For each variable give: value, confidence (high, medium, low), and for
  B1_5, B5_2, B5_3, B5_4 a one-line reason citing paragraph numbers.
- Never quote more than ten words from the article.

Variables
[paste the definition, values and rule of each admitted variable from the codebook]

Answer only in the JSON format required.
```

Matching JSON schema (excerpt)

```
{
  "type": "object",
  "properties": {
    "id": {"type": "string"},
    "codes": {
      "type": "object",
      "properties": {
        "B1_8": {"type": "object",
                 "properties": {"value": {"type": "integer", "enum": [1, 2, 3, 4, 99]},
                                "confidence": {"type": "string", "enum": ["high", "medium",
    "low"]}},
                 "required": ["value", "confidence"], "additionalProperties": false},
        "B5_2": {"type": "object",
                 "properties": {"value": {"type": "integer", "enum": [1,2,3,4,5,6,7,8,9,10,99]},
                                "confidence": {"type": "string", "enum": ["high", "medium", "low"]},
                                "reason": {"type": "string"}},
                 "required": ["value", "confidence", "reason"], "additionalProperties": false}
      },
      "required": ["B1_8", "B5_2"], "additionalProperties": false
    }
  },
  "required": ["id", "codes"], "additionalProperties": false
}
```

RESTRICTED DISTRIBUTION

Restricted distribution · FLL members only · Do not copy or circulate

Appendix C. Checklists by phase

Phase	Done when
1 Corpus	Sampling rule written; zone rules checked on test pages; logs reviewed; failures retried; corpus frozen and counted
2 Exploration	KWIC, keyness and actor tables saved; articles flagged for close reading
3 Codebook	Every code has definition, values, rule and anchor; version frozen; pilot findings decided
4 Training	60 articles coded; memos written; codebook memos reviewed
5 Reliability	Alpha per variable reported; weak variables revised and retested
6 Claude	Prompt and schema frozen; validation run twice; variables admitted; audit done
7 Analysis	One data set; B8 computed; RQ scripts run from scratch without error
8 Writing	Every claim has a number and a passage; quotations verified; methods section complete

Appendix D. Glossary

Term	Meaning
Anchor example	A real passage that clearly fits one value of a code
Contra-flow	Media flow from outside the Western center (Thussu, 2007)
Document variable	A value attached to a whole document in MAXQDA
Frame	A pattern of problem, cause, evaluation and remedy (Entman, 1993)
Keyness	How much more often a word appears in one group of texts than in the rest
KWIC	Keyword in context: each use of a word with its surroundings
Krippendorff's alpha	Agreement between coders, corrected for chance
Link environment	The links around an article, by zone and rank
Page track	Screenshots, rendered page and links captured with each article
Same-event set	Articles from different outlets on the same event and day
Sponsor state	The state that funds or owns an outlet
Structured output	API answers that follow a fixed JSON schema

RESTRICTED DISTRIBUTION

Restricted distribution · FLL members only · Do not copy or circulate

Appendix E. References

Entman, R. M. (1993). Framing: Toward clarification of a fractured paradigm. *Journal of Communication, 43*(4), 51–58.

Fairclough, N. (1992). *Discourse and social change*. Polity.

Galtung, J. (1998). High road, low road: Charting the course for peace journalism. *Track Two, 7*(4).

Galtung, J., & Ruge, M. H. (1965). The structure of foreign news. *Journal of Peace Research, 2*(1), 64–91.

Hayes, A. F., & Krippendorff, K. (2007). Answering the call for a standard reliability measure for coding data. *Communication Methods and Measures, 1*(1), 77–89.

Karlsson, M., & Sjøvaag, H. (2016). Content analysis and online news: Epistemologies of analysing the ephemeral Web. *Digital Journalism, 4*(1), 177–192.

Krippendorff, K. (2018). *Content analysis: An introduction to its methodology* (4th ed.). Sage.

Kuckartz, U., & Rädiker, S. (2019). *Analyzing qualitative data with MAXQDA*. Springer.

Lynch, J., & McGoldrick, A. (2005). *Peace journalism*. Hawthorn Press.

Miskimmon, A., O'Loughlin, B., & Roselle, L. (2013). *Strategic narratives: Communication power and the new world order*. Routledge.

Rodriguez, L., & Dimitrova, D. V. (2011). The levels of visual framing. *Journal of Visual Literacy, 30*(1), 48–65.

Thussu, D. K. (Ed.). (2007). *Media on the move: Global flow and contra-flow*. Routledge.

Welbers, K., van Atteveldt, W., & Benoit, K. (2017). Text analysis in R. *Communication Methods and Measures, 11*(4), 245–265.

Check page ranges and editions against the originals before submission.

Software documentation: MAXQDA user manual (maxqda.com/help); quanteda (quanteda.io); Claude API documentation on structured outputs and batch processing (platform.claude.com/docs).

RESTRICTED DISTRIBUTION

Restricted distribution · FLL members only · Do not copy or circulate