Shifting to supervised topic assignment

Following informal consultation (September 18, 2026)

Created September 24, 2026
Updated November 15, 2026

Table of contents

Arguments for the shift 1
Objections and responses 4
Coding design 8
Framing in the paper 11
Recommendation 11
Appendix: Initial codebook 11

> **Note**
>
> I wrote this document in September 2026, after an informal consultation on how to code the emails, and revised it alongside the mockup through October 22. The restructuring plan has since superseded it. The plan codes rhetoric, functions and actions as three kinds of codes, and starts coding without a codebook.

The shift replaces NMF topic assignment at a 15% threshold with human coding of passages. A coder selects a passage of newly written content in a purpose-built page and gives it one topic code. An email has no topic codes of its own, and the analysis phase derives its topics from its coded passages. Coders code a sample of 100 to 200 emails from the pilot corpus this way, and those codes guide and validate a local model that applies them to the full corpus. The pilot's four themes carry over as categories of the codebook, alongside an Other category.

Arguments for the shift

1. Fixed categories let the expanded corpus test the pilot

The protocol describes the pilot as a pipeline test that produced preliminary observations, and the expansion to the Scientific Director's and data managers' mailboxes exists to find out whether those observations hold at scale. Refitting

NMF on a corpus with different senders and a different vocabulary yields a new solution with its own k and its own topic labels, so a pilot topic such as Serology and Data Harmonization has no guarantee of reappearing. A comparison across the two corpora would then compare two different sets of categories. Supervised codes hold the categories constant, so the expanded corpus can retest, on the same constructs, the finding that agreements work and data operations work ran as parallel workstreams.

NMF stays in the methods as the exploratory stage that generated the category scheme, and the codes do not guide its fitting. Its dominant-topic assignments stratify the annotation sample so that it includes rare categories, and agreement between each email's dominant NMF topic and its dominant coded theme reports how much of the pilot's structure the coded labels reproduce. Because the codes did not shape the topics, that agreement is evidence.

2. Human labels supply validity evidence that coherence scores cannot

The c_v score measures whether a topic's top words co-occur. It says nothing about whether a topic corresponds to a type of administrative work. The two least coherent topics, Meeting Scheduling and Invitations (0.473) and Executive Correspondence (0.497), make up the entire Coordination theme, whose generic language the codebook has to define most carefully. The LLM-generated labels also needed correction by hand. The application and the protocol label one topic Catalogue Completion Reminders; the preprint relabels it Catalogue Onboarding and Collaboration after manual inspection showed it captures early introductions to the Maelstrom catalogue. Five of the 14 labels differ between the protocol's Table 1 and the preprint. A label attached after the fact is a hypothesis about what a word cluster means, and every downstream analysis treats it as a measurement.

A supervised design reports validity as coder agreement on the double-coded sample and as precision and recall for each theme against the human codes.

3. The 15% threshold measures vocabulary mix

A weight threshold on a normalized NMF matrix records how an email's vocabulary divides across topics. In the pilot, 72.7% of emails cleared 15% on two or more topics (mean 2.01 per email). Four of the Agreements topics (T0, T5, T7, T8) share near-identical subject lines and separate on body vocabulary. The preprint reads them as stages of the agreement lifecycle, and that reading carries weight because four of the cluster's five most strongly coupled pairs (25 to 33%) are among them. The model cannot tell a change in activity from a change in wording, so the reading remains an interpretation. Coding tests it. Annotators either distinguish the stages reliably or they do not, and reliability is reportable.

The threshold is also a free parameter. The protocol weighed 10%, 15%, and 20%, and every phase, role, gender, and co-occurrence result inherits the choice. A coder deciding which topic a passage carries replaces the threshold with a

judgment that the team can document and check for reliability, and an email's split across topics follows from where each topic appears in its text.

4. Passage codes restore single counts and record where topics arise

Under the threshold scheme an email appears in every topic cell it clears, so the cells in the phase, role, and gender figures overlap. The application specified a simpler design in which, for the workflow phase and sender analyses, "emails will be assigned to their dominant topic", with the multi-topic threshold reserved for co-occurrence. The preprint applied the 15% threshold to every analysis.

Passage coding restores the application's design. Each passage takes one topic, and an email's profile is the share of its coded words in each theme, with word count as the weight because it is the only quantity a coder's decision directly produces. Each email counts once in the phase, role, gender, and network analyses, with its count divided across themes in proportion to those shares, and the theme with the largest share supplies the dominant topic that the application specified. The profile has no cap, so the 25.1% of pilot emails that carried three or more topics keep every topic. Co-occurrence describes the themes that appear together in an email, and the position of each passage also records which topic arises first. Because a coder assigns each topic to a stretch of text, the profile reflects what the email does in its own words, in units that sum to the email's coded length.

5. One coded construct serves the topic, network, and questionnaire analyses

A single paper needs one unit of analysis across its topic and network sections, and the coded email is that unit. With network analysis folded in, topic labels become attributes of edges. In the person projection each edge inherits the topics of the emails exchanged, and at 2.01 topics per email those inherited labels are overlapping sets, which complicates claims about which topics carry bridging ties, such as data managers linking study teams to the Harmonization Group on Data Operations. The derived profile gives each email a weight of one divided across themes, so an edge's topic profile is a distribution over themes that sums to one.

In the email co-participation projection, nodes that carry several topics tie topics together through the labels themselves as well as through shared people, which blurs the line between that projection and the co-occurrence analysis. Labelling each node with its dominant theme restores the line, with co-occurrence describing content through passage-level topics and the projection describing people.

The codebook's role categories supply node attributes with definitions fixed in advance. The four exclusion flags give the protocol's exclusion of systemic notification traffic a coded basis that the analysis can report and audit.

The questionnaire's second and third domains correspond to the email analysis's topic and phase categories, and its thematic coding starts from the email analysis framework. Defined codes with inclusion rules move into that coding intact. An NMF topic moves as a list of keywords.

6. A local model already in the pipeline scales the codes

Manual annotation covers a sample of 100 to 200 emails from the pilot corpus, and a local model applies the codes to the remaining emails and to the expanded corpus, which grows well beyond the pilot's 6,833 emails. The pipeline already runs Meta-Llama-3.1-8B-Instruct on secure servers to label topics, so few-shot prompting with annotated examples extends existing infrastructure and keeps email content on McGill servers. A sample of this size supports prompting and validation, and the codebook's anchor examples serve as the few-shot examples, so annotators and model work from one artefact. Passage-level codes also give the model the exact text behind each code. The annotation itself is qualitative coding, the competence the questionnaire analysis also requires, and it follows the process used for the BibVik codebook.

Objections and responses

The first three objections are serious. The design handles the rest.

1. The codebook inherits the pilot's structure, so confirmation is partly circular

The top-level categories are the pilot's four themes and the example codes come from its topics. Coders applying those definitions will tend to sort emails as the topic model did, so the design partly builds in agreement with the pilot's findings. A fixed codebook also cannot show what the pilot missed. The annotated sample comes from the operations manager's mailbox, so the validation covers that mailbox only, and the Scientific Director's mailbox will carry governance and leadership correspondence the sample never covered.

Response. Add an Other category for substantive passages that fit no theme, review the passages coded into it after annotation, and version the codebook with dated changes. Annotate a further small sample from each added mailbox before applying the codes to it. Fit NMF to the expanded corpus as well and report any topic that no code accounts for. Discovery stays in the design, and the confirmatory claims rest on categories that could change.

2. The threshold and single-count arguments do not require supervision

An unsupervised design can fix the overlap problem. The application already specifies dominant-topic assignment for phase and sender analyses, and a threshold sensitivity analysis, a lower k, or merging the four Agreements topics would

address the rest. A reviewer can accept everything said about the 15% threshold and about single counting and still ask why humans need to label anything. Fixed categories, validity evidence, defined codes shared with the questionnaire and the network roles, and local classification are the arguments that justify supervision itself.

Response. Lead the justification with fixed categories and validity evidence, and present the single count and the record of where topics arise as additional benefits of coding.

3. A sample of 100 to 200 emails cannot support fine-grained claims

The codebook lists example codes under four themes and an Other category, and the full set will be larger. Spread across 100 to 200 emails, rare codes receive a handful of examples or none, so per-code precision and recall are unstable. A proportion estimated on 100 to 200 emails carries a 95% interval of about 7 to 10 percentage points at its widest. Fine-tuning a classifier on this few labels is out of reach, and few-shot prompting is the workable route. The main analyses also report per-study medians and drop cells with fewer than five emails, so misclassification in rare codes moves results more than it would in pooled counts. LLM output varies with temperature and seed, and the pilot already ran its LLM tasks across five temperature settings to check stability.

Response. Make the four themes the validated classification targets and keep the finer codes as descriptive annotation. Report agreement and per-theme precision and recall with bootstrap confidence intervals. Run the model at temperature 0 with fixed seeds and limit claims to themes that meet a stated performance floor. Propagate the confusion matrix into the main results by resampling assigned labels, and report how far the headline findings move.

4. Passage-level codes have no reliability record

Agreement on passages needs coders to agree on which stretches of text carry a topic, where each stretch begins and ends, and which topic it carries. Two coders can choose the same code for slightly different stretches, split one stretch into two passages, or leave a stretch uncoded that the other codes. A single topic per passage removes the disagreement over which of several codes applies and adds a disagreement over where to split. The pilot's human review rested mostly on one research assistant. A sample of 100 to 200 makes double-coding feasible, which removes most of the capacity problem.

Response. Double-code the whole sample and adjudicate disagreements. Measure agreement in three places. For the derived email profiles, use Krippendorff's alpha on each email's dominant theme, with 0.67 as the floor for tentative conclusions and 0.80 for firm ones. For passages, count two coders' passages as the same when they overlap on more than half their words, report the share of passages matched, and report alpha for the code assigned to matched passages against the same floors. Because passage length is the profile's weight, also report

the difference in word count between the two coders' versions of each matched passage, since two coders can agree on a passage and its topic while disagreeing on its length by enough to move the email's profile. Selections snap to whole words, which removes the commonest source of that difference. Since the model must also find passages, evaluate it at the sentence level by projecting human passages onto sentences. If passage agreement fails and email-level agreement holds, analyze at the email level using derived profiles alone. If both fail, keep NMF with dominant-topic assignment.

5. A single topic per passage forces a choice

Emails often do several things in one sentence, such as returning a signed agreement and proposing a call. A single topic per passage forces the coder to split the sentence or to drop one of the topics, and either choice loses information that the threshold scheme kept as overlap.

The constraint follows from the weighting. Passage length is the profile's weight, and each word counts once, toward the theme of its passage. A passage carrying two topics would need its length counted toward each, which inflates the email's total past its coded length, or a further rule to split its length between them, which brings back a ranked judgment under the name of a topic assignment. A single topic per passage keeps the weighting well defined, so an email's coded length divides across themes with no remainder and no word counted twice.

Response. Write a splitting rule into the codebook. Code at the level of the clause where possible, and where one clause carries two topics, code the topic the sender asks the recipient to act on, or the main information the email conveys when it requests nothing, and start a new passage for the other. Report the share of passages coders mark unsure, and run the main results without them as a check.

6. Creating codes while coding erodes the codebook

Coders can create a code from the selection when no existing code fits. Each new code changes the codebook mid-round, so emails coded earlier never had the chance to receive it, and two coders can create overlapping codes for the same thing. Free creation is also the quickest route to a codebook that drifts from its definitions.

Response. Allow creation only during calibration. A code created on the fly is provisional, records the email it came from, and stays undefined until a coder writes a definition in the codebook drawer. The team reviews provisional codes after each calibration round, merges duplicates, and recodes affected passages before the codebook freezes. Once it freezes, the create option turns off, and a coder who finds a gap codes the passage in the Other category.

7. The codes could guide the topic model directly

Seeding or regularizing NMF with the codebook, or fitting a guided model to the annotated emails, would put the codes to work on the topics themselves. A sample of 100 to 200 emails is too small to supervise a factorization across 14 or more topics. Guidance also makes the topics reflect the codebook. Agreement between NMF and the human codes then becomes partly built in, a fresh NMF on the expanded corpus loses its value as a check on what the codebook misses, and something the codebook produced replaces the pilot's unsupervised results.

Response. Keep NMF unguided and use the codes to evaluate it. Report agreement between each email's dominant NMF topic and its dominant coded theme, weigh it alongside c_v when comparing k and stopword choices, choose those on one split of the sample, and report final agreement on the held-out split. Fit a fresh NMF to the passages coded in the Other category, and treat topics that surface there as candidates for new codes. Other serves as the intake point for the codebook's next revision.

8. Dropping actions loses the record of what an email asks for

An earlier draft of the codebook coded actions such as request, provide, chase, confirm, and decide alongside topics. The intent was a relation that could sit between a topic and the party it concerned, moving toward a subject-predicate-object structure in which the action is the verb. That structure needs a well-formed object, such as what a request asks for or what a chase pursues. No part of the codebook identifies that object consistently, so coders coded the verb with nothing for it to attach to, and an action code alone cannot say what the sender requested or from whom. Retrofitting the object later, from role codes or from passage adjacency, would add to the codebook without drawing on anything the design already produces.

Response. Drop actions from the codebook, since a verb with no object adds a code without the information it was meant to carry. If the analysis later needs the coordination-versus-friction distinction that actions aimed to capture, design the object slot first, for example by linking a passage to the role codes of its sender and recipient, and add the verb once the codebook specifies the relation it completes.

9. The change departs from the application and the preprint

The application names NMF, its evaluation against three alternatives, and LLM-generated labels as the method, and the preprint reports NMF results. Dropping NMF as the primary method can read as abandoning the validated pipeline. The preprint also presents NMF with LLM annotation as a replicable approach other groups can adapt, whereas supervised coding asks each group to build a codebook for its own corpus.

Response. Keep NMF as the exploratory stage, record the change in the

protocol with its rationale and date, and leave the preprint unchanged as the pilot record. Publish the codebook and annotation guidelines so the codebook becomes the transferable output.

10. Annotation widens human exposure to email content

Presidio as configured replaces names, email addresses, and URLs, so phone numbers and institutional and study details remain in the text. Annotators read each sampled email in full. The sample size of 100 to 200 keeps that exposure small, and the reading shows which identifying details de-identification missed.

Response. Coders flag any passage that contains identifying information, marking the passage whether or not it also carries a topic. The team redacts flagged passages before quoting or publishing them, and before using them as few-shot examples. The flag rate also audits the de-identification step. Confirm that the ethics file (REB #26-08-029) covers annotation by the research team, and annotate only cleaned, de-identified text on the secure servers, under the confidentiality terms that apply to the pipeline research assistant.

Coding design

A mockup of the coding page, built with invented emails, shows the design this section describes.

Code from a selection

Coding happens on the fly. A coder selects a passage in the newly written content, and a popover opens beside it. The coder picks a category, which is one of the four themes or Other, and a search box lists that category's existing codes and offers to create a new one when none matches. The passage takes that one code. The number keys pick a category, so the mouse stays in the text. Each coding carries its own unsure flag, and each passage carries its own flag for identifying information, which can stand alone on a passage that has no topic.

The interface keeps three levels apart. Passage codes attach to selected text and show as highlights coloured by category, with a wavy red underline for identifying information. Each email's Flags section holds the email-level flags and the note. Roles belong to people, and coders set them in the review pane. When a passage already has a code, the popover shows it with a Remove button, and the review pane offers the same control along with the unsure and identifying-information checkboxes.

Passages do not overlap. Selections snap outward to whole words, and arrow buttons move either edge of a passage by one word, so a loose selection does not become a boundary disagreement.

Leave email-level topics to the analysis phase

Emails carry no topic codes of their own, and the coding page shows no email-level profile. The analysis phase derives the profile from the passages, which gives each email a single measurement of its topics that cannot disagree with them. The profile is each theme's share of the email's coded words, chosen because passage length is the only quantity a coder's decision produces, and a one-word passage should not weigh as much as a paragraph. Coders code every substantive passage of newly written content and leave greetings, sign-offs, and quoted history uncoded, so the coded length reflects content and excludes message boilerplate. Recompute the profile from passage counts as a sensitivity check. Coders flag an email with no substantive content instead of coding it.

Separate topic from actor

The example codes mix subjects (Informed consent), activities (Cleaning, Uploading), and actors (Contracts officers, Research ethics boards). Contracts officer also appears as an Institutional role, so coders would code the same information twice. Topic codes cover activity and subject matter, and actors go to the role coding.

Define codes and fix the unit of coding

Each code gets a definition, inclusion and exclusion criteria, and two or three anchor examples, including a boundary case. Written rules cover the adjacent codes that coders will confuse, such as Schema / map / spreadsheet / variables against Data dictionary / metadata, Harmonization against Core data elements, and Data governance against Data sharing agreements. The pilot's strongest cross-theme coupling was between variable mapping and the data dictionary, so that boundary gets its rule first. Coders code what the sender does in the newly written content of the email. They may see the prior message for context and ignore quoted history, so coders code a reply that quotes a long agreement thread for its own content.

Every code anchors to the passage that justifies it, and a passage takes one topic. The anchoring limits over-application, speeds adjudication, and allows a check on whether the model cites the same passages. The splitting rule in the codebook says where one passage ends and the next begins.

Add flags

Six flags carry what the codes do not. Four apply to an email in place of coding it, and exactly one of the four applies to a given email. Too short covers content too brief to code, such as a one-line acknowledgement. Delivery status notification covers a mailer-daemon or bounce message, Out of office an automatic vacation reply, and Automated notification any other system-generated message, such as a calendar invite or a platform alert. Pruning removes content-free emails from

the sample before coding begins, so these flags catch what the pruning missed, and a high rate of use signals that the pruning step needs revisiting. Setting any of the four clears any passages already coded on that email, so a flagged email carries no orphaned codes and the flag is the complete record of why it was not coded. The analysis reports agreement on these email-level flags with the rest, and their counts alongside the coded corpus.

Two further flags apply to a passage or a coding, and neither constrains the other. Identifying information marks a passage that contains details de-identification missed. It can stand alone on a passage with no topic or sit alongside one, since the flag and the code record different things about the same text. Removing it from a passage with no code also removes the passage, because a passage exists only to hold a code, a flag, or both. Unsure marks a single coding that the coder doubts, since doubt attaches to a coding decision. Content that fits no theme takes a code in the Other category. A note field holds anything the second coder or adjudicator should know. The flags feed codebook revision, and the reliability analysis reports agreement with and without the unsure codings.

Code roles on the person, kept simple for round 1

Round 1 codes each person once per study from a single list of roles grouped by domain, chosen from a select box in the review pane, with the date the role applies from. A crosswalk from the pilot's five sender roles to the codebook roles keeps pilot findings comparable. Point man becomes Point person, since gender is an analytic variable in this study. Automated services remains a role, and emails from it take one of the four exclusion flags, most often Automated notification.

Roles are the least settled part of the codebook. A role depends on the circumstances of an action and an interaction, and a title conflates it with position. A list of positions does not capture what a person did in a given exchange. Round 1 does not settle this. The disagreements coders show on roles decide afterward whether roles stay positional, become situated in the email or the interaction, or split into both. Coding people and institutions mentioned in the text waits on that decision.

Stage the process and keep the codebook as data

Coders code two calibration rounds of about 20 to 30 emails independently and discuss them, and those emails stay outside the sample. The team freezes the codebook version before the double-coded sample begins, and a dated changelog records later changes. Codes created on the fly during calibration stay provisional until reviewed, and creation turns off at the freeze. Annotators do not see NMF topics or model predictions, since those anchor judgments and undermine the independence of the NMF comparison. The sample uses a floor of about 25 emails per theme, with the theme taken from the NMF dominant topic, and a cap on emails per study so high-volume studies do not dominate. Coordination needs

deliberate oversampling to reach that floor, since it is the smallest theme in the pilot, with about 1,600 topic assignments against about 3,800 for Agreements.

Codes, definitions, and examples live in a YAML file that renders into the qmd and feeds the coding page and the model prompt, so the three cannot drift apart. This matches the literate programming commitment in the protocol.

Framing in the paper

Present the design as two stages within the email work package. NMF with LLM labelling is the exploratory stage that produced the category scheme from the pilot corpus. Hand coding of a 100 to 200 email sample is the confirmatory stage that fixes those categories, and agreement between the local model and the human codes validates the model's labels at scale. NMF stays unguided by the codes, so the two stages remain independent checks on each other. The same codes then carry into the network and questionnaire analyses. The application names NMF as the method, so the change goes into the protocol as a documented refinement, with the pilot's topic model retained as the comparison showing how much of its structure the coded labels reproduce.

Recommendation

Run two calibration rounds on emails outside the sample, review the codes created on the fly, and freeze the codebook version. Then double-code the 100 to 200 email sample and treat its agreement thresholds as the go/no-go decision. If the sample meets them, apply the codes at the theme level with a few-shot local model that uses the codebook's anchor examples and that the sample validates, keep the Other category, and refit NMF on the expanded corpus without guidance from the codes. If they are not met, keep NMF with dominant-topic assignment, which is the design the application already describes.

Appendix: Initial codebook

Overview

This is a list of potential codes and code categories developed as part of a plan to shift toward a supervised method. I draw from the preliminary findings of the pilot stage.

The initial set of primary code categories are the same as the themes identified in the pilot study, and are subject to further modification or expansion. I include a few exemplary codes for each code category, which were inspired by the topics and subtopics from each theme. I will also include a brief description of each category.

This is similar to the process for developing the codebook for the BibVik project: https://zackbatist.info/BibVik/codebook

I also document some early thinking on role definitions below.

Codebook

Topics

Agreement and contracts

- Data governance
- Research ethics boards
- Data sharing agreements
- Contracts officers
- Informed consent
- Amendments / extensions / waivers

Data operations

- Transfer
- Cleaning
- Preparation
- Uploading
- Schema / map / spreadsheet / variables
- Core data elements
- Harmonization

Study documentation

- Data dictionary / metadata
- Cohort description
- Publications
- References / citations
- Authorship / credit
- Open data
- Institutional repository

Coordination

- Meetings
- Introductions
- Contact info
- Calls for participation

Roles

Roles are situated within three domains: CITF Core, Data Contributors, Institutional. These domains align with the project's general goal, which assumes a distinction between the CITF Databank and its data contributors that is bridged through operational processes.

As of right now (September 22, 2026), the institutional domain is still tentative. The other two domains comprise communities of practice with assumed knowledge and ways of working, which are attempting to align themselves. The institutional domain may be oriented by a set of priorities that may be generalized across institutions, but the roles within the domain do not necessarily comprise a cohesive unit. If the CITF core and each data contributor team are bounded collectives, then the institutional domain comprises relatively loosely-associated individual actors who interject with their distinct yet generalizable perspectives.

CITF core

- Leadership
- Data managers
- Operations manager
- Project liaison

Data contributors

- Project lead
- Point person
- Data manager

Institutional

- Contracts officer
- Information technology support
- Administrative contact
- Automated services