THE FRAME

Statement Normalizer

Making the implicit structure of prescriptive claims visible, testable, and debatable.

GRDprocess Sàrl — March 2026

www.nextinsight.org/normalizer

What Does the Statement Normalizer Do?

Decompose

Any prescriptive statement ("we should…", "governments must…") is broken into 5 structural elements: subject, verb, object, justification, expected outcome.

Evaluate

Each element is tested for unbound variables — information the reader needs but doesn't have. Elements are classified as explicit (complete) or ambiguous (with named questions).

Iterate

The user corrects ambiguous elements. The system re-analyzes with the same structural tests. Each iteration makes the claim more precise — and the hidden assumptions more visible.

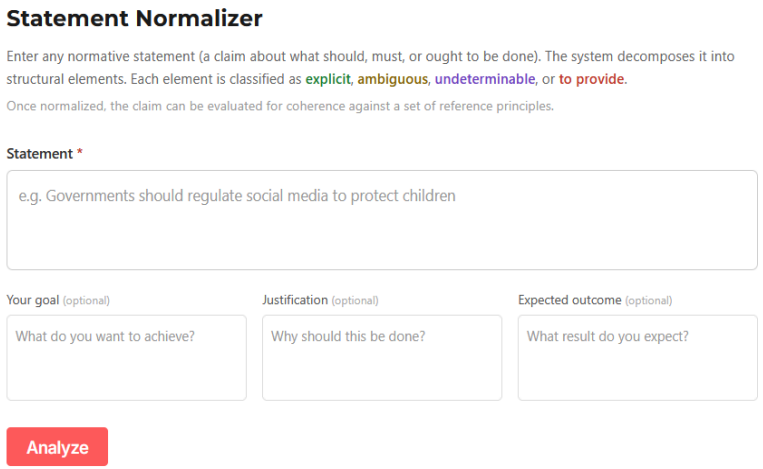
Output → Analyzer

Once all elements are explicit, the normalized claim becomes the structured input for THE FRAME Analyzer — which evaluates coherence between principles. Normalization must precede analysis.

The tool does not judge if a statement is correct. It reveals what is implicit, assumed, or undefined.

The Interface

Normalizer



Input fields

Statement
The prescriptive claim to normalize

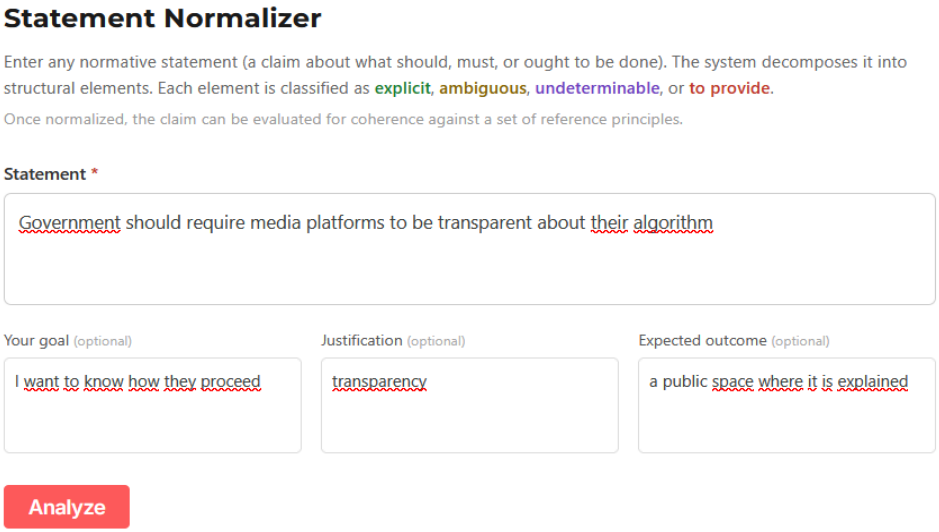
Your goal
Always explicit — only you can state it

Justification
Why should this be done?

Expected outcome
What concrete result do you expect?

Live prototype: www.nextinsight.org/normalizer

Example: Entering a Normative Claim



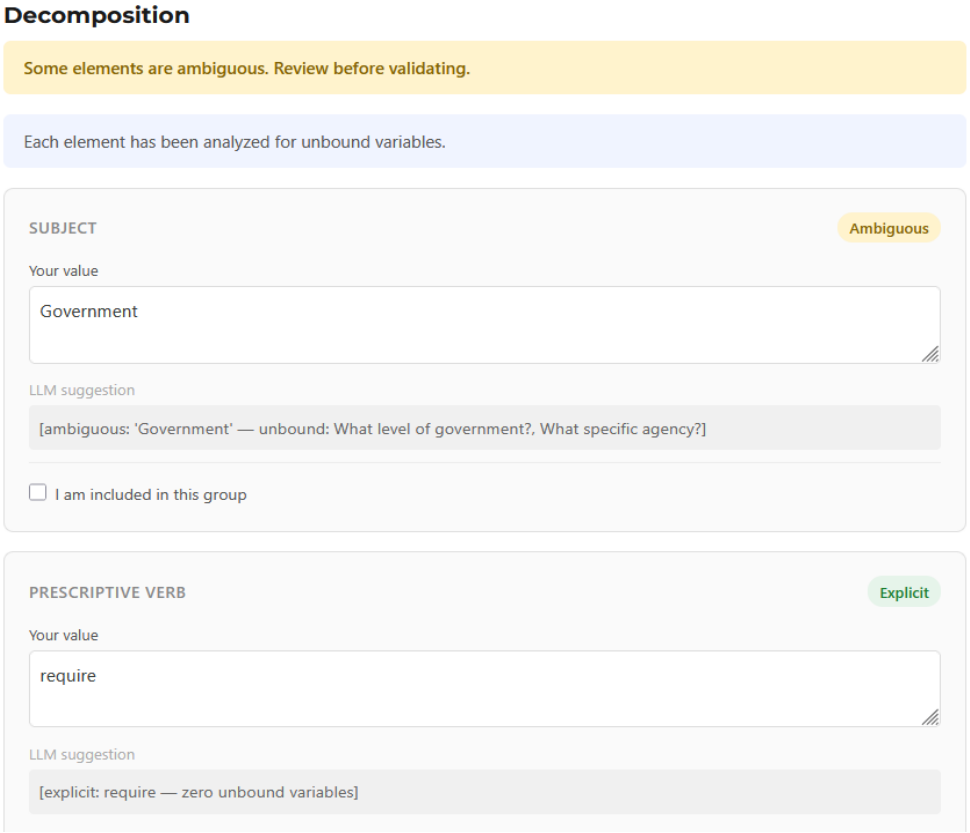
User input:

"Government should require media platforms to be transparent about their algorithm"

The user also provides a goal, justification, and expected outcome. These are optional but help the LLM produce a richer decomposition.

The system will now decompose this into structural elements and test each one.

First Analysis: Decomposition Results



"Government"

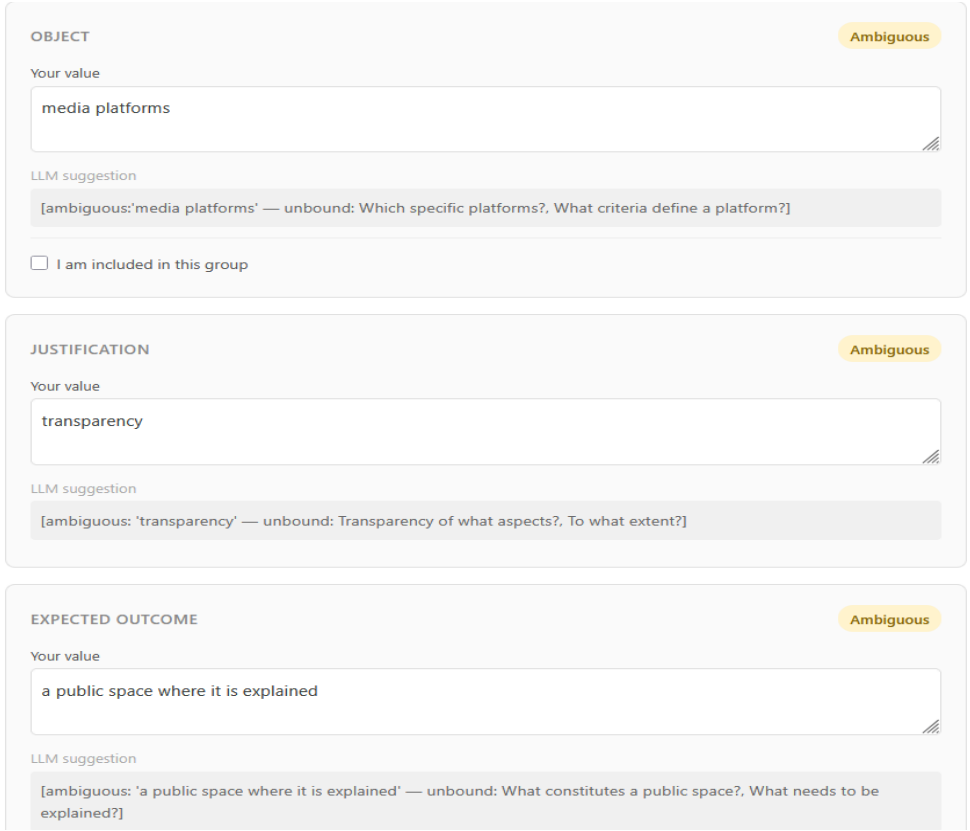
Ambiguous — the LLM identifies unbound variables: What level of government? What specific agency?

"require"

Explicit — zero unbound variables. The prescriptive verb is structurally complete.

The LLM does not suggest answers — it names the missing information as questions.

First Analysis: Remaining Elements



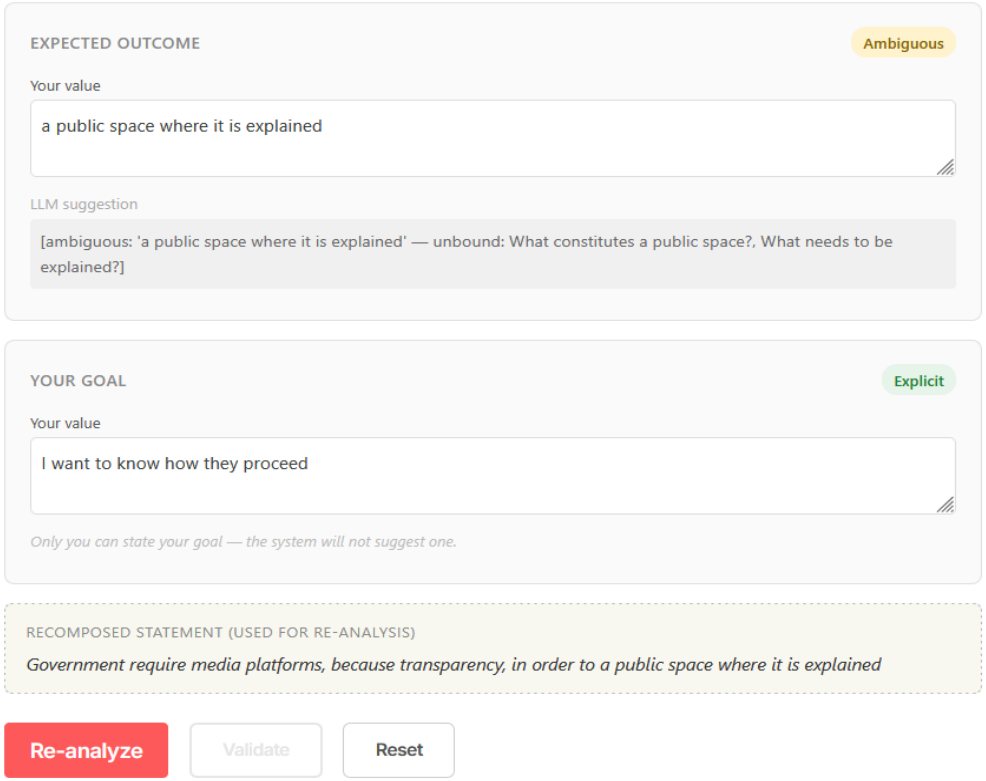
"media platforms"
Ambiguous — Which specific platforms? What criteria define a platform?

"transparency"
Ambiguous — Transparency of what aspects? To what extent? An abstract concept with unsaturated relational arguments.

"a public space where it is explained"
Ambiguous — What constitutes a public space? What needs to be explained?

Every ambiguity is a structural gap in the claim that was previously invisible.

Goal & Recomposed Statement



Your Goal: always explicit

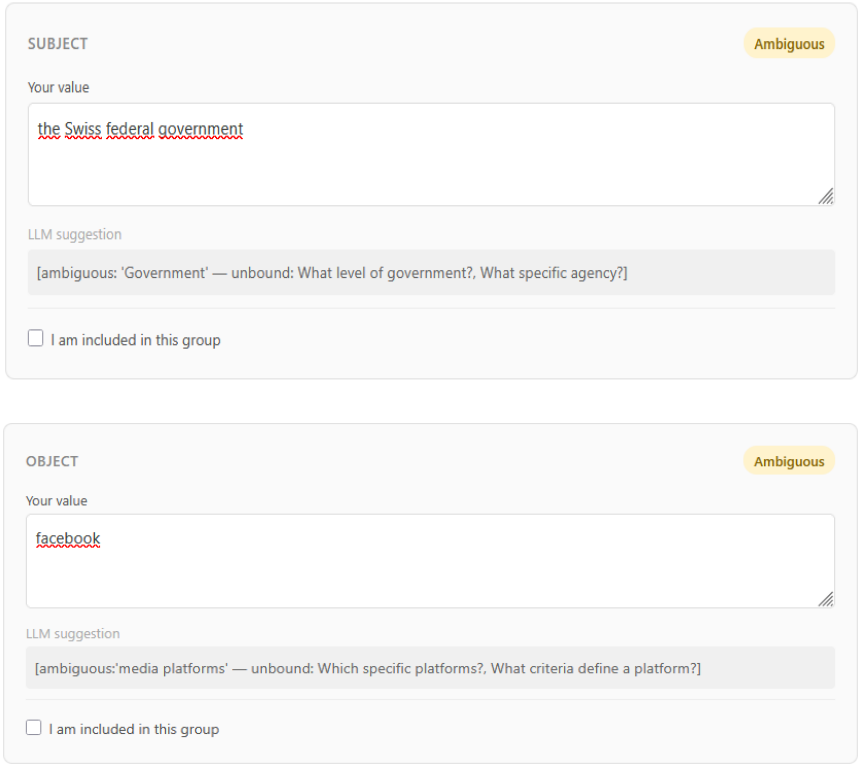
The system never evaluates the user's goal — only the user can state it. It will be used later for coherence analysis.

Recomposed Statement

The system generates a recomposed version from the current elements. This is the version sent for re-analysis — making the normalization process transparent and traceable.

The user can now modify any ambiguous field and trigger a re-analysis.

Iteration: User Corrects Ambiguous Elements



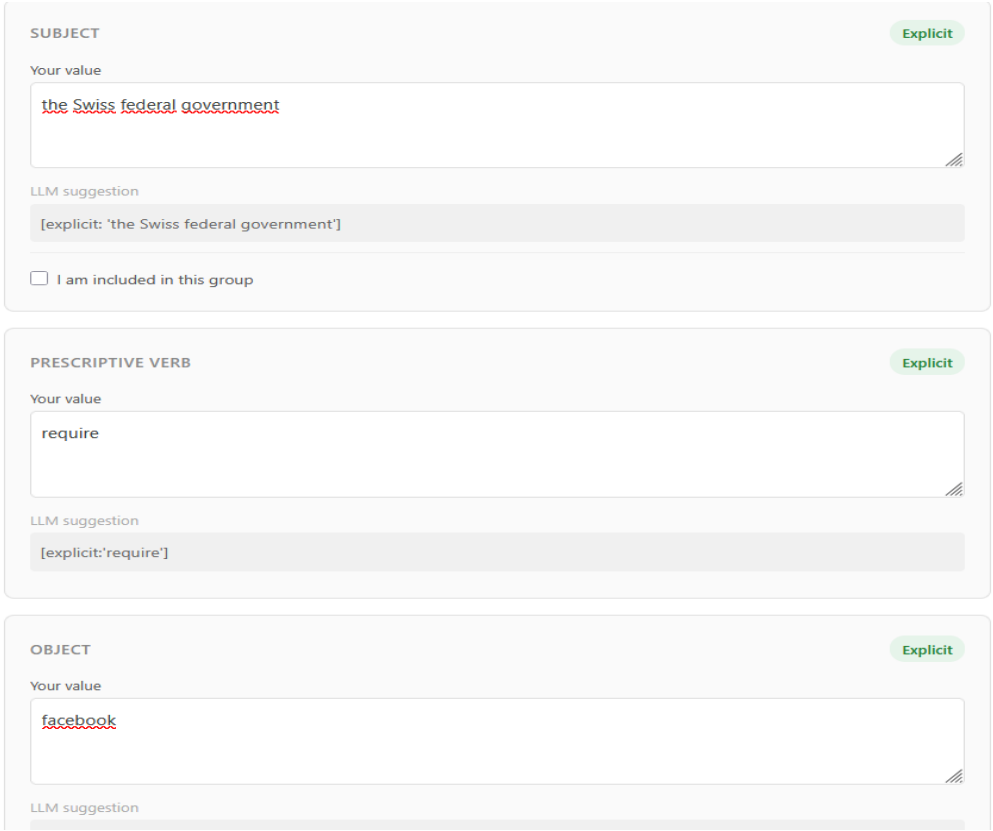
The user replaces vague terms with precise values:

"Government" → "the Swiss federal government"
Resolves the referential variable: which government?

"media platforms" → "facebook"
Resolves the referential variable: which platform?

The LLM suggestion still shows the previous analysis. The user types a correction and hits Re-analyze.

After Re-Analysis: Elements Become Explicit



Subject: explicit

"the Swiss federal government" — singular identifiable entity, zero unbound variables.

Verb: explicit

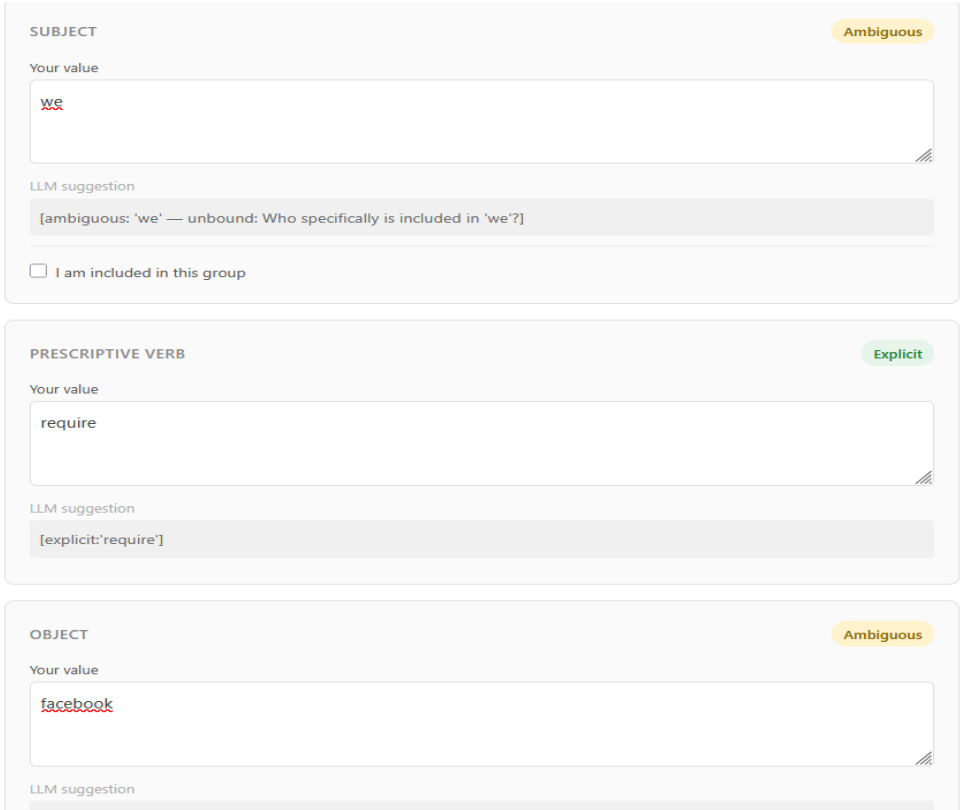
"require" — structurally complete action.

Object: explicit

"facebook" — proper noun, single entity.

The structural tests work: precise terms pass, vague terms don't.

Regression Test: What Happens If We Go Backwards?



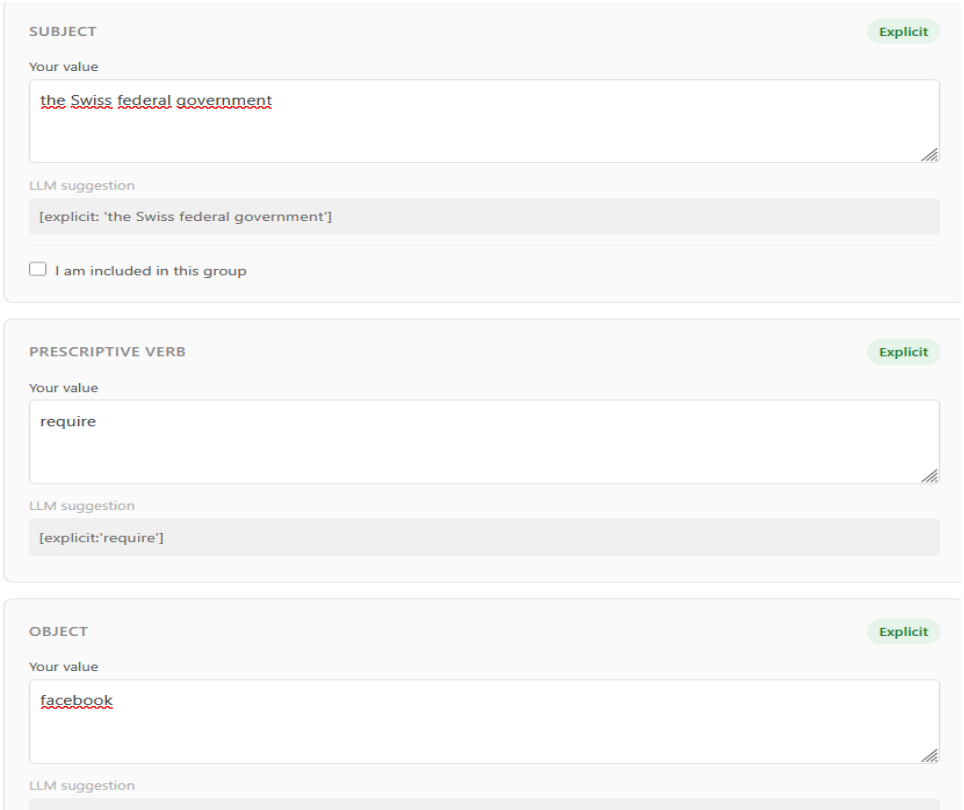
User replaces "Swiss federal government" with "we"

Subject immediately returns to ambiguous: "Who specifically is included in 'we'?"

Interesting: facebook also returns to ambiguous. When the subject is vague, the scope of the object becomes uncertain too.

This demonstrates contextual reasoning — the LLM evaluates elements in relation to each other, not in isolation.

Stability: Restoring Precise Terms

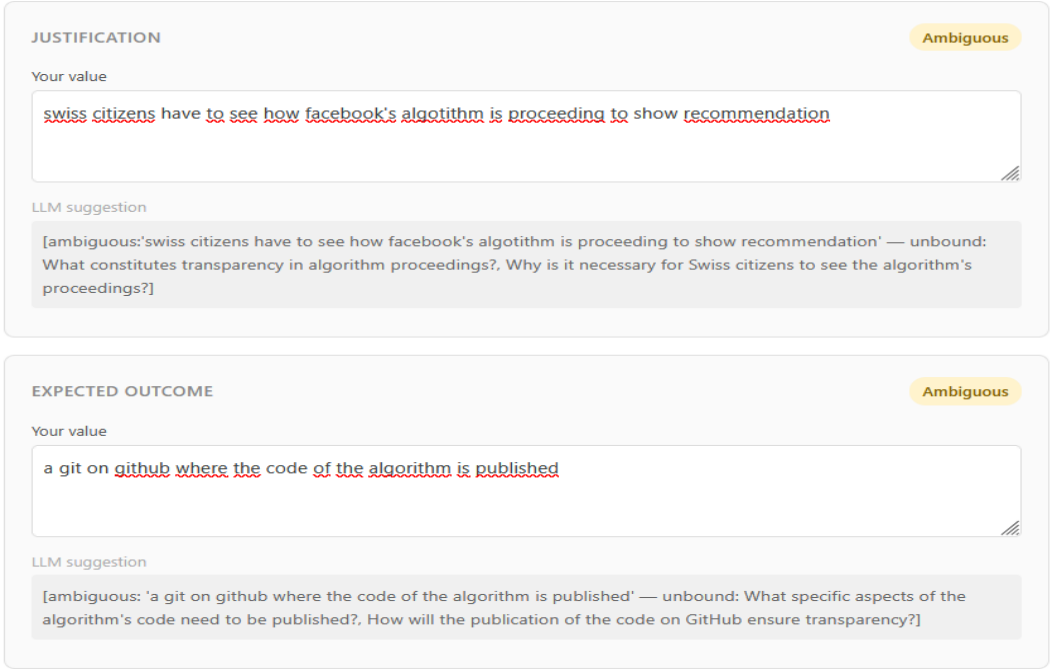


User restores "the Swiss federal government"

Subject, verb, and object all return to explicit. The system is stable — same inputs produce the same outputs.

This consistency is critical for a normalization tool. Previous model (Mistral-24B) produced different results on identical inputs between iterations.

Challenge: The LLM Asks Too Many Questions



Second-level questions

The LLM sometimes asks questions about the presuppositions of the terms rather than the terms themselves:

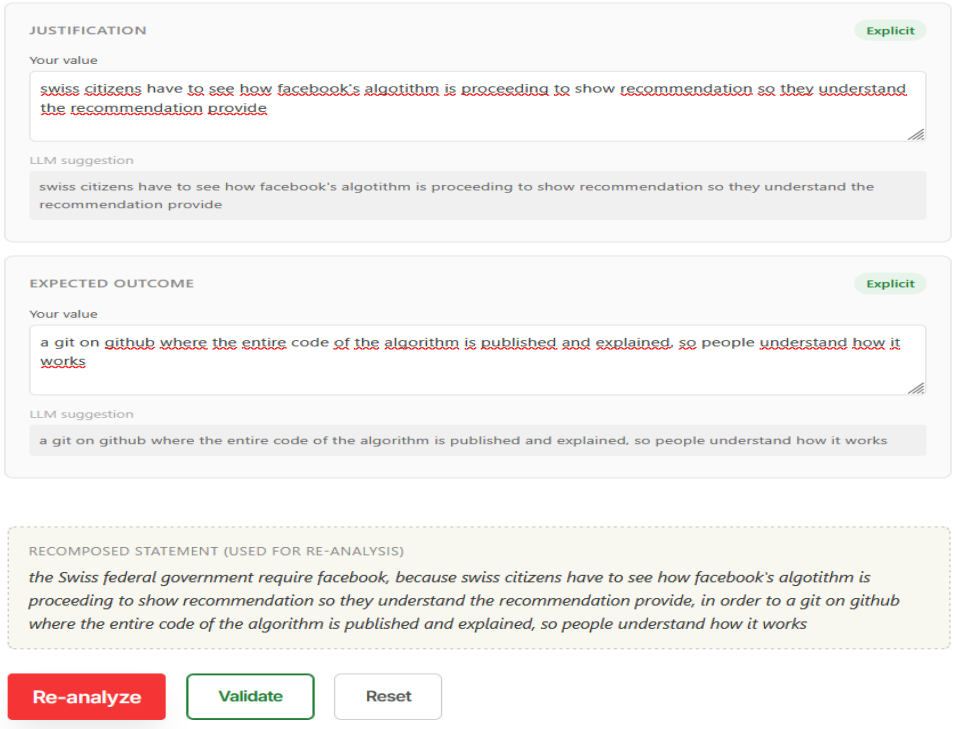
"Why is it necessary for Swiss citizens to see the algorithm?"

This is a meta-justification of the justification — circular and outside scope.

Planned feature:

A "closed" status where the user accepts an ambiguous field. Pending questions stay visible as metadata — contestable in debate, but not blocking.

Multiple Passes → All Elements Explicit



After further refinement:

Justification and expected outcome both pass to explicit. The Validate button becomes active.

Normalization flow:
Vague claim → structural decomposition → ambiguity detection → user correction → re-analysis → all explicit → validated.

Each step is traceable. Each decision is documented.

Validated Normalization: Structured Output

Validated Normalization

ORIGINAL STATEMENT

Government should require media platforms to be transparent about their algorithm

SUBJECT — Explicit

the Swiss federal government

Skin in the game: Not included

PRESCRIPTIVE VERB — Explicit

require

OBJECT — Explicit

facebook

Skin in the game: Not included

JUSTIFICATION — Explicit

swiss citizens have to see how facebook's algotithm is proceeding to show recommendation so they understand the recommendation provide

EXPECTED OUTCOME — Explicit

a git on github where the entire code of the algorithm is published and explained, so people understand how it works

YOUR GOAL — Explicit

I want to know how they proceed

Iterations: 12

Send by email | Back To Edit | Reset

The final output:

Every element has an explicit status. The original vague statement has been transformed into a fully specified, testable claim.

This structured output (subject, verb, object, justification, outcome, goal — all explicit) is the input for the Analyzer, which will evaluate coherence between competing principles.

12 iterations

It took 12 iterations to reach all-explicit. This reflects the depth of hidden assumptions in a single normative claim. A planned "closed" status will let users stop earlier when precision is sufficient for their context.

The user can send the result by email or return to edit.

What's Next

"Closed" Status

Let users accept ambiguous fields as sufficient for their context. Questions stay visible as contestable metadata.

Dedicated Reasoning Agent

Combine a deterministic module (structural rules) with LLM (semantic questions) for consistent analysis.

THE FRAME Analyzer

Evaluate coherence between normalized principles using a 15×15 relationship matrix (support / tension / opposition).

AI Safety Application

Apply the Normalizer to AI constitutional documents (Anthropic, OpenAI) to reveal structural ambiguities in alignment principles.