From Corpus to Argument:

Classification, Evidence Retrieval, and Obligated Composition

Flyxion
Independent Researcher

October 2026

Abstract

A corpus-based writing system must distinguish organizing existing material from proposing and developing a new argument. This essay specifies a staged architecture in which classification organizes a corpus, retrieval assembles source passages, recombination proposes an argumentative connection, and Yarnwinder develops the proposal under recorded obligations. Four independent organizations describe subject matter, discovered similarity, processes, and the task-relative roles of passages, and the roles are treated as operations applied at retrieval time and not as owners of passages. Successive prompts to one local language model propose connections, examine their premises, recover qualifications and identify discriminating cases, each receiving the outputs before it. A coordinating stage determines whether a proposal is ready for drafting, requires further evidence, or remains unresolved. Acceptance concerns accurate source interpretation, an explicit connection, adequate treatment of objections, and an inventory of open claims. The essay also records what a review of earlier implementations found, namely that the architecture is more substantial than the evidence that the code worked, and it lists the specific defects to repair. Entropy and embedding similarity remain descriptive diagnostics. The architecture is a design specification and not a report of demonstrated performance.

1 The Separation of Operations

A writing system can produce fluent prose while concealing the difference between a source claim and a conjecture introduced during composition. The present design makes that difference persistent. Classification describes existing material. Retrieval selects passages relevant to a task. Recombination proposes a relation among selected works. Composition develops that relation into an argument whose premises and limitations remain available for examination.

The output of each operation becomes an input to the next, but no operation inherits evidential authority merely by occupying a later position. A passage does not become reliable because it belongs to a cluster, and a proposed connection does not become established because it has been expanded into several sections.

2 Four Organizations of the Corpus

The Micromegas ontology supplies an explicit, overlapping classification. Whales concerns living beings on Earth. Mushrooms concerns rapid growth and propagation. Trees concerns rootedness or spatial fixation. Lamphron concerns dark and baryonic matter. Ocular Xylem concerns photonic computers combined with polyphase building materials based on wood pulp. Lamphrodyne concerns dark energy associated with expanding voids and inflaton fields. These definitions are supplied categories for the project, and their use does not establish the physical claims of an essay classified under them. Voltaire's story of the same name is a literary reference and is kept apart from the definitions: a passage is not assigned to a category because it resembles the story, and each classification records which version of the definitions was applied.

Assignments depend on argumentative relevance. A biological subject does not require a Lamphron assignment merely because it consists of baryonic matter. Each assignment records whether the category names a literal subject, an explanatory role, or an analogy, together with the passage supporting that interpretation.

Topic discovery supplies a second organization through similarities among passages. Its clusters are provisional descriptions of the corpus and not authoritative subjects into which every work must fit. An essay can participate in several clusters, and an outlying passage can remain unassigned. Cluster names are interpretations attached after inspection of representative material, and the record keeps the embedding model and the clustering run that produced each grouping.

A process ontology supplies a third organization. It describes transformations such as inscription, accumulation, compression, selection, routing, verification, commitment, exclusion, repair, delegation and continuation. The vocabulary is provisional and expected to change as the corpus is read more closely. The classification distinguishes a process described by an essay from an argumentative operation performed by the essay, because describing compression and compressing an argument are different activities.

The fourth organization concerns task-relative roles. The Archivist retrieves provenance, alternatives and qualifications. The Formalist retrieves definitions, dependencies and constraints. The Synthesist identifies proposed connections. The Strategist examines practical conditions and consequences. The Curator coordinates these operations. The five roles come with stated objectives and stated failure modes, and the failure modes are the useful part: the Archivist risks hoarding context, the Formalist risks correct but ungrounded abstraction, the Synthesist risks never converging, the Strategist risks trading understanding for immediate effect, and the Curator risks spending its effort on coordination. A passage can serve several roles, and its usefulness can change with the question. No role owns a passage. Each role is a query that can be put to any set of passages, and role tags, where stored, are an index recording who judged a passage useful for which role, when, and on what instruction. The tags accelerate retrieval and are open to revision. They carry no authority over the passage, which suits a corpus whose essays are retained whatever any role thinks of them.

3 What the Earlier Implementations Establish

The design draws on a long record of exchanges in which a language model proposed scripts for iterative essay writing, a five-role memory architecture, and an extensive scaffold of schemas, tests, user interfaces and container files. The record establishes the intentions with considerable clarity. A working summarizer already existed, which walks a directory tree, splits each text file into chunks, sends each chunk to a local Granite model through Ollama, appends the summaries to a file, and logs progress so that interrupted runs resume. The proposed essay writer extends this by generating an outline, expanding each heading, scoring the result with a second model call and rewriting until a threshold is met. The proposed memory architecture assigns incoming passages to one of four personas under a fifth that coordinates.

The record does not establish that the implementations work. Several exchanges describe the scaffolds as complete, validated or production-ready, and the repository summary accompanying one of them states in its own notes that the model calls are stubbed and the embedding projection is a placeholder. A scaffold with stubbed inference is a structure into which an implementation could be placed, and calling it validated conflates the existence of a test file with the passing of a test against real behavior. Where fuller implementations of a component appeared earlier in the record, later versions sometimes replaced them with shorter ones. The appropriate use of this material is therefore to recover the design intentions and treat the code as text to be inspected, which Section 9 does.

4 Retrieval Before Recombination

A draw of two or three works supplies a candidate combination, using the seeded recombiner described elsewhere with one of five relations: transfer, tension, composition, boundary or, for three works, triangulation. Retrieval then assembles an evidence packet containing passages, their locations, their source versions, and enough surrounding context to interpret them. A summary can guide retrieval, but source-dependent judgments require access to the relevant source passages.

The evidence packet records its omissions. An excerpt may depend on a definition outside its boundaries, qualify a claim in a later section, or refer to an external result that has not been checked. These conditions remain visible and do not disappear into a general statement that the source has been consulted.

Selection strategies can use any of the four organizations. A draw might connect similar topics with different processes, or a shared process across different Micromegas categories. Uniform random selection remains a comparison condition. No strategy is presumed to improve argumentative yield before evaluation.

5 Successive Roles on One Model

The proposed implementation uses successive prompts to one local Granite model through Ollama. Distinct prompts express distinct tasks, and their succession does not constitute independent corroboration. Five simultaneously running models are not required.

The Synthesist proposes a connection that requires every selected work and identifies what each contributes, together with the premise the connection needs and the excerpts do not supply. The Formalist examines the premises needed to make that connection, separating implications, analogies and new conjectures, and lists the definitions the connection depends on, where each is stated, and any place where one word carries different meanings. The Archivist then makes a second retrieval pass, now that a connection exists to be tested. It searches the classified corpus and not only the drawn works, selecting by Micromegas category, topic cluster and process term in addition to embedding similarity, and reports for each passage whether it supports, limits or contradicts the connection. The first retrieval gathers evidence about the drawn works, and this second pass gathers evidence about the proposal. The Strategist develops a concrete case, implementation condition or discriminating investigation through which the connection might become answerable, and states what result would count against it.

Each later stage receives the evidence packet and the relevant earlier outputs. Repeating an unchanged prompt without supplying earlier responses is repeated generation and not deliberation. Any iterative exchange therefore records what changed, which objection prompted the change, and which obligations remain.

6 The Curator's Limited Decision

The Curator assigns one of three dispositions. A proposal is ready for drafting when its connection is explicit and its evidential requirements are sufficiently identified to permit development. It needs passages when an identifiable omission could materially change its interpretation, in which case the Archivist pass is repeated with a narrower query. It remains an unresolved question when the proposed connection cannot presently support the intended argument. The third outcome is a legitimate product, since a combination whose bridge cannot be supported is information about the combination, and the record of why it failed is kept with its address.

Readiness authorizes composition and does not certify correctness. A proposal may be worth drafting precisely because it develops a clearly identified conjecture. Conversely, an attractive proposal may remain unresolved because its connecting premise cannot yet be stated adequately.

The decision includes a rationale and the outstanding obligations. When a bounded deliberation budget is exhausted, the system records that fact. It does not relabel exhaustion as confidence or force a single interpretation merely to complete the run.

7 Composition Under Recorded Obligations

Yarnwinder develops the approved brief into an outline, section drafts and supporting appendices. Each section retains its intended contribution, source passages, introduced conjectures, objections and outstanding obligations. A revision creates a new version and does not overwrite the record needed to understand earlier decisions. Development proceeds in order: for each section the Archivist and Formalist queries are run against the brief's passages and the classified corpus, the model drafts the section, the Strategist query proposes a concrete case, and the obligations are checked and recorded before the next section begins.

Acceptance rests on four obligations. Source interpretations must be accurate within the scope claimed. The connecting premise must be explained and not supplied through shared vocabulary. Material objections must receive an answer, a concession, or a corresponding restriction of scope. Open claims must remain explicitly listed. The ledger records each obligation for each unit as discharged with a pointer to what discharges it, deferred with a stated reason, or open. A section is accepted when no obligation is open, and a deferred obligation is carried to the section where it will be discharged or to the final list of verification items. A model may propose that an obligation is discharged and cite the supporting passage, and a reader confirms or rejects the proposal. The ledger keeps proposals separate from confirmations, so that the essay is not credited with checks that only the model performed on its own output.

A section can therefore be acceptable as a conjectural argument without its conjecture being established. The relevant question is whether the prose represents that status accurately. An objection that defeats a central claim requires revision or withdrawal of that claim, and a stylistically improved restatement does not meet it.

Section-level acceptance does not establish whole-essay adequacy. The integrated draft requires a further examination of dependencies, terminological consistency, duplicated arguments, incompatible assumptions, and qualifications lost between sections. Citation inspection likewise distinguishes detecting a reference from verifying that it exists and supports the associated claim.

8 Diagnostics Without Certification

Unigram Shannon entropy describes a token-frequency distribution. It is unchanged by any permutation of the same tokens and therefore cannot certify coherence. A short test confirms the point: a sentence of thirteen words and a shuffled copy of the same thirteen words both have entropy of about 1.925 nats. The blending formula proposed in the earlier record is worse than uninformative, because it converts entropy to a score by subtracting ten times the entropy from one hundred and capping the result, so that lower entropy always yields a higher score. That rewards repetition although the accompanying text claims the measure penalizes it, and a section repeating a few words would be favored over a varied one.

Embedding similarity describes proximity under a specified representation and cannot cer-

tify preservation of a claim, since a small wording change can reverse a proposition. A high similarity between successive revisions shows that little changed, and a low similarity shows that much did, and neither shows whether the change was an improvement. Routing weights express assigned relevance to a task and not calibrated probabilities of truth. In the earlier script they were produced by adding fixed increments when a role's output contained particular fields, so a weight records whether an output had a certain shape, and a threshold test that treats it as confidence attaches a meaning the number lacks. Model-generated evaluations are recorded assessments whose reliability remains to be investigated. A model scoring text that the same model wrote measures its agreement with itself, and a failed parse that defaults to zero, as in one proposed evaluator, would trigger rewriting for reasons unrelated to quality.

These measures can identify material for inspection. Lexical entropy can flag unusually repetitive or dispersed text, and embedding similarity can serve as a drift alarm for a revision that departs far from the text it was meant to repair. They do not determine acceptance.

9 Defects to Repair in the Supplied Code

A reading of the scripts finds specific problems. The JSON extraction helper uses the pattern `(\{(?:[^{}]|(?1))*\})`, which relies on a recursive group reference. Python's standard `re` module does not support it, and compiling the pattern raises an error reporting an unknown extension. The third-party `regex` package supports recursion, but the standard-library-only constraint that the rest of the pipeline observes is better served by scanning for an opening brace and calling `json.JSONDecoder.raw_decode` from each candidate position until one parses.

The curator script invokes the persona script with the argument `-` and the chunk on standard input, while the persona script opens its second argument as a file path, so it attempts to open a file literally named `-`. The persona script also passes the chunk twice, once embedded in the prompt and once on standard input, and builds a shell command by string interpolation with `shell=True`, which in one version quotes the prompt and in another does not. Passing an argument list without a shell, with the chunk supplied once, removes the conflict.

The debate loop runs up to three rounds in which every persona receives the same prompt and the same chunk as in the previous round, so no round has access to any other round's output. The rounds differ only through sampling variation, and the confidence test that ends the loop is applied to weights computed from heuristics on field presence. The loop does not debate, and a debate requires that later prompts include earlier responses, as Section 5 specifies.

Further changes follow from the earlier sections. The stubs in the scaffold must be replaced by real calls or labeled as stubs wherever the repository is described. The tests must exercise the model path or be described as exercising only the routing heuristics. Every script should report a failed model call as a failure and not convert it to a default score. None of these repairs has been carried out, and the specification is written so that they can be made independently.

10 Storage and Evaluation

The extension `.memory` remains a convention for plain-text memory material that a person can read and edit with ordinary tools. Machine-generated classifications, source locations, model outputs, version records and processing status are stored separately, and derived indexes can be rebuilt without rewriting the source notes. The classification index holds, for each passage identifier, its source file and hash, its Micromegas assignments with the kind of assignment and the evidence passage, its topic cluster with the embedding and clustering versions, its process terms, and its role tags with their provenance. The processing record holds, for each run, the seed, the corpus snapshot, the combination address, the relation, the model identifier, the version of each prompt, and every stage output in order.

Evaluation uses a fixed corpus snapshot and a pre-specified sample. It records source accuracy, whether every selected work contributes substantively, the explicitness of the connection, the treatment of objections, and the completeness of open-claim inventories. Additional observations include retrieval omissions, processing cost, revision failures and human review burden. It is not known how often a drawn combination yields a brief the Curator would accept, how often a reader confirms the obligations a model proposes as discharged, whether five role prompts to one Granite model produce output that differs usefully from a single prompt, or whether the Micromegas categories, discovered topics and process terms help retrieval beyond what embedding similarity alone provides. Each of these questions can be examined by running the pipeline on the sample and having readers judge the outputs against the full sources, and none should be settled by assumption.

Generated briefs remain outside the source pool. Essays developed from them enter only through an explicit review and accession procedure. This separation permits investigation of the generator without allowing its own outputs silently to redefine the evidence from which later proposals are drawn.

11 Conclusion

The pipeline divides the work so that each part claims only what it can support. Classification organizes the corpus without deciding what an essay should say. Retrieval assembles evidence without deciding whether it suffices. Recombination proposes an argument and says plainly where the sources fall short of it. Development expands an accepted proposal and records, at every step, what it rests on and what is still owed. The roles survive as questions that can be put to any set of passages, and the ledger of obligations replaces the numerical scores that the earlier designs asked of a model. The earlier record leaves a clear architecture and a body of code that has to be inspected and repaired before it can be relied on, and the first step is to carry out the repairs listed above on a small corpus and to read what comes out.

A Minimum Section Record

A section record identifies the section and its version, its role in the argument, and the exact source versions and passage locations used. It preserves the drafting instruction, model identifier, response, subsequent assessments and revision history. It distinguishes source claims from introduced inferences and conjectures.

The record also states the disposition of each material objection and lists unresolved verification obligations. Acceptance names the assessment procedure and the scope of what was accepted. Compilation success, stylistic fluency and a favorable model score are recorded separately from argumentative acceptance.