# Attest: A Claim-Native Data Layer for Shared Knowledge

Fedor Chishchin
Omic, Inc.
fedor@omic.ai

Steve Muller
Omic, Inc.
steve@omic.ai

Ivan Meić
Omic, Inc.
ivan@omic.ai

Buğra Sarı
Omic, Inc.
bugra@omic.ai

Olamide Isreal
Omic, Inc.
olamide@omic.ai

Gabriel Richman
Omic, Inc.
gabe@omic.ai

## ABSTRACT

Organizations share documents rather than facts, so every reader re-derives what a corpus says. Attest is a claim-native data layer that stores one row per source assertion and uses two content hashes, one for the proposition and one for each source's assertion of it, as the storage layer's grouping key. Each ingredient exists in prior provenance models; this vision paper examines their combination as the write target of a language-model extraction pipeline. On 150 biomedical concept pairs with opposed SemMedDB predications and a median of 2,960 candidate statements (in a 20-pair reading, 14 oppositions were extraction errors, so the task tests attribution rather than disagreement), answers built from the store's records cited no record from outside the pair (0 of 1,148), against 32 of 4,156 for source-tagged top-130 retrieval and 323 of 2,754 for reading the whole pool. Ablations attribute most of the gain to indexing assertions by concept pair, and part to an opposition vocabulary that the benchmark's answer key shares. On a 50-paper fixture a vector index was cheaper and recovered more figures, and on 24 questions whose disagreeing papers human annotators had labelled, retrieving twenty abstracts beat the store at half the prompt (0.65 against 0.40). Two extraction builds of the same corpus shared 18% of their propositions; propositions asserted by more papers were more often rebuilt (30%, 83% and 94% for one, two and three or more sources), an association accounted for by a higher per-source re-extraction rate and more opportunities. These preliminary results suggest that a shared claim store can be copied and replayed reliably but not yet independently re-derived.

## 1 INTRODUCTION

An organization that wants its members to work from shared knowledge gives them shared documents, and what is derived from those documents stays private. Each reader extracts what the documents state, weighs it, notices or misses disagreement between sources, and keeps the result in a form others cannot query. Two analysts reading one corpus produce two summaries that cannot be compared, merged or audited, because the summaries share no unit of identity.

Retrieval systems automate this reading without changing its structure. Retrieval-augmented generation [14, 29] and long-context models derive an answer at read time, once per question, and discard the derivation. Repeated or concurrent questions can return different answers with no record of whether the difference arose in the question, the retrieval or the model.

A shared record must answer four questions about every fact it holds: who asserted it, when, how strongly it should be believed, and whether it still stands. Document stores answer none of them at the storage layer. Graph models can represent them: an RDF graph is a set, so a bare triple stated twice is one triple [10], but named graphs, singleton properties, RDF 1.2 reifiers and property-graph edges can each keep one source's statement separate [2, 7, 19, 39]. Common practice often attaches sources to one statement instead: Wikidata accumulates references on a single statement [44, 45]. Nanopublications give each assertion a verifiable, content-derived identifier and record retraction and supersession as data [25, 26]; micropublications model support and challenge between claims [9]; and recent work attaches multi-source support and conflict to nanopublications [33]. Provenance vocabularies and system-versioned tables record derivation and history [27, 28].

**Related work.** Web-scale knowledge bases have long been built by extraction over corpora [6, 11, 36, 43], and weak supervision made labelling cheap enough to be routine [42]. Databases have a mature theory of where a value came from [5, 8, 17, 20] and mature machinery for versioning it [18, 22, 32]. Language-model extractors raise the quality ceiling [21, 23, 38, 41] and make the write path non-deterministic [3].

Attest is a claim-native data layer that combines these mechanisms in one storage design and measures it under a language-model write path. This vision paper contributes:

1. **An integrated record** (§2) that stores one row per source assertion and uses two content hashes, one for the proposition and one for each source's assertion of it, as the storage layer's grouping key, so that counting sources and finding potentially contested propositions are group-bys. Each ingredient exists separately in the systems above; the contribution is their combination as the write target of an extraction pipeline.
2. **Preliminary measurements of what that design buys and costs** (§3). On 150 biomedical questions over opposed SemMedDB predications, most of them extraction errors in a 20-pair reading, attribution from the store's evidence is close to that of source-tagged top-130 retrieval at a similar prompt size, and ablations locate the difference in selecting the records on the two opposed predicates, which the benchmark's answer key also uses. A vector index is cheaper and recovers more figures on a 50-paper fixture, and on 24 questions whose disagreeing papers were labelled by human annotators, retrieving the documents beats the store.
3. **A characterization of write-path instability** (§4). Two builds of one 50-paper corpus share 18% of their propositions; agreement is higher for propositions asserted by more papers, and that association is accounted for by both

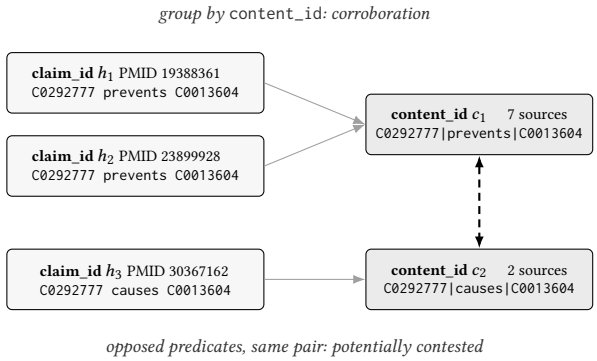


**Figure 1: Assertions and propositions for one SemMedDB concept pair. Each source's assertion is its own row (`claim_id`); grouping by `content_id` counts sources per proposition, and a declared opposition pair marks the two propositions for review. Three of nine records shown.**

a higher per-source re-extraction rate and more opportunities.

## 2 A CLAIM-NATIVE RECORD

### 2.1 The claim record

Attest stores one row per assertion and records the four answers of §1 as fields:

```
(subject, predicate, object,
 confidence, provenance, time, status)
```

Operations that a conventional store applies *to* rows are written as claims, with their own sources and times, in the same table as all other claims.

- *These two names denote one thing* is a `same_as` claim. Equivalence classes are maintained by union-find as such claims arrive; a `not_same_as` claim splits a class.
- *This source was withdrawn* is a `retracted` claim. Affected claims receive a status change rather than deletion, so what was believed before the withdrawal, and on whose authority, remains queryable.

Single and batch writes extend a SHA-256 chain over claim identifiers, so a reordered or altered write sequence is detectable against a trusted copy of the chain. The chain does not yet cover bulk loads or status changes, and it records identifiers rather than claim bodies.

### 2.2 Two identities: proposition and assertion

Each claim carries two hashes.

```
content_id = H(subj | pred | obj)
claim_id   = H(subj | pred | obj | source
               | type | time [| payload])
```

`content_id` identifies the *proposition*; `claim_id` identifies *one source asserting it at one time*, and includes a digest of the claim's payload when it has one. Two papers reporting the same finding yield two `claim_id`s and one `content_id`. Entity names are normalized and resolved before hashing, so `content_id` depends on the state of identity resolution at write time.

Using `content_id` as the grouping key makes corroboration a group-by over stored rows (Figure 1). Predicates declared as opposites in the vocabulary (`activates`/`inhibits`, `causes`/`prevents`) let a second group-by find *potentially* contested propositions: claims whose subjects and objects match while their predicates oppose. Whether such a pair is a genuine disagreement depends on context the triple does not carry. Qualifiers such as population, dose or comparator are not part of `content_id`; they enter `claim_id` through the payload, so assertions made under different conditions share a proposition and must be read with their conditions.

None of these elements is new on its own (§1). What Attest adds is their use as the storage layer's grouping key under a language-model write path: each source's assertion stays a separate row, so aggregating a proposition does not discard which source holds which side.

Confidence is derived from source type, the number of distinct sources on a proposition, and age, with a default 365-day half-life. Because age is computed at read time, two readers obtain the same value only from the same snapshot at the same evaluation time.

### 2.3 Identity as the critical stage

With `content_id` as the grouping key, every downstream operation inherits the quality of entity canonicalization [15]. Corroboration miscounts when two spellings of one entity are stored as two entities, and ranking and aggregation read the same equivalence classes. A claim-native store therefore concentrates in one stage a risk that a document store distributes across its readers.

Entity matching conventionally decides, mention by mention, whether one string resembles another closely enough to denote the same entity, by blocking and scoring [40] or by a trained model [31, 37]. Many of those decisions do not need a model. In nine identity passes over text corpora, 149 of 264 model merges joined orthographic variants such as a plural and its singular, and the model treated such variants inconsistently, merging `left knees` with `left knee` while refusing `falls` and `fall`. Attest therefore settles orthographic variants with a normalizer and reserves the model for deciding whether two different names denote one thing.

For those decisions Attest applies the claim record to its own vocabulary. It records, for each term a corpus uses, what a document states the term means, with a source and a date:

```
Reading = (definition, source_id, time)
```

A definition is then an assertion like any other: matching readings corroborate, and differing readings of one name are candidates for equivocation. Supplying recorded definitions to the identity arbiter reduced missed merges from 15 to 9 over 492 decisions without changing the number of wrong merges (6). The equivocation check is implemented but flagged none of 27 terms on a seven-paper corpus, including `ADF`, which denotes two regimens there, because one paper states both definitions.

## 3 PRELIMINARY RESULTS

### 3.1 Attribution of opposed predications

The benchmark takes 150 pairs of biomedical concepts for which disjoint sets of PubMed records assert opposed SemMedDB predications [24] (`causes`/`prevents` 97, `activates`/`inhibits` 53). Pairs

**Table 1: Attribution on 150 opposed-predication pairs, median pool 2,960 statements. Minority: mean share of the minority side's records cited. Exact: every contested record cited and none from outside the pair. Wrong: cited records that assert nothing about the pair, over all answers. †Ablations: the store's records for the pair, or for its two opposed predicates only, as flat sourced lines.**

| | tokens | minority | exact | wrong / cited |
|---|---|---|---|---|
| top-8 retrieval | **402** | 0.49 | 0.19 | 3 / 908 |
| top-130 retrieval | 4,349 | 0.81 | 0.57 | 32 / 4,156 |
| whole pool | 86,040 | 0.59 | 0.01 | 323 / 2,754 |
| claim store | 5,415 | 0.88 | 0.48 | **0 / 1,148** |
| pair records† | 1,963 | 0.96 | 0.77 | 2 / 4,435 |
| opposed records† | 951 | **0.99** | **0.85** | 1 / 2,843 |

were drawn from the most-cited opposed pairs and stratified by the ratio between the sides; the minority side has a median of two records. The gold sides are the predications themselves, defined by the same opposition vocabulary the store uses. They are not verified disagreements: reading one record from each side of 20 sampled pairs against its abstract (a model reader, not a human) found 14 pairs holding an extraction error, usually on the minority side, 4 genuine context-dependent effects and 2 concepts too generic to disagree about. What follows therefore measures whether a reader reports which record asserts which predicate, not whether it surfaces disagreement; §3.5 asks the second question. Each pair's pool holds every other claim on either concept, a median of 2,960 statements, each tagged with its source record. A single model, Gemini 3.1 Flash-Lite at temperature 0, answers every arm under one prompt, and answers are scored against record identifiers without a model judge (Table 1).

Reading the whole pool fails: shown every record, the reader cites 0.59 of the minority side and 12% of its citations concern other pairs. Source-tagged top-130 retrieval at a similar prompt size does far better. Against it, the store's answer context cites more of the minority side (+0.07, 95% CI +0.01 to +0.14) and no record from outside the pair (0 of 1,148; below 0.3% at 95% confidence), but recovers the exact set no more often (-0.09, CI -0.19 to +0.01), because it adds graph neighbours and summaries to the contested records.

The ablations locate the effect. Showing only the store's records for the concept pair, flat with their sources, raises minority recall to 0.96 and exact recovery to 0.77 in a third of the tokens; restricting them to the two opposed predicates reaches 0.99 and 0.85. Grouping the same records by predicate changes neither, while layouts that print a source count before the identifiers lead the reader to omit identifiers in about half its answers. The gain comes from indexing each source's assertion by concept pair, which any schema keeping one row per source supports, and the further step from opposition is aligned with the benchmark, whose answer key uses the same vocabulary. The store and ablation arms are given the pair's concept identifiers; on the 22 questions that do not name them, top-130 retrieval falls to 0.45. Repeated answers at temperature 0 were identical, so the intervals reflect items, with pairs sharing a concept resampled together. With pools small enough to read whole, about 130 statements, a reader given every sourced statement matched the store. [`attribution_scaled`, `presentation_controls`]

**Table 2: Eight questions over a 50-paper corpus. The vector index retrieves eight passages and the claim store sixty claims. Build and per-answer costs in US dollars at list price, including output tokens; the full-corpus row is an input-token estimate, not a run. s: seconds per answer; figures: gold values recovered out of 16, range over three runs; papers: mean distinct papers cited on the four contested questions.**

| | build | per answer | s | figures | papers |
|---|---|---|---|---|---|
| full corpus | **none** | $0.045 | – | – | – |
| vector index | $0.009 | **$0.0007** | **7.3** | **15/16** | 2.6 |
| claim store | $1.39 | $0.0028 | 15.7 | 10–13/16 | **7.5** |

## 3.2 The cost of structuring

Structuring a corpus into claims costs more than indexing it. Table 2 compares three ways of answering the same eight questions over one 50-paper corpus (1.97M characters, 454K tokens). The claim store is the 50-paper build of §4, which ran extraction and no identity resolution. Every answer is produced by `gpt-5.6-luna` at temperature 1.0 under one system prompt; the retrieval rows are means over three runs. [`read_cost_50`]

**A vector index is cheaper on every cost axis.** It builds for $0.009 in 14 s against $1.39 of extraction for the store, answers at a quarter of the price in under half the time (3,209 prompt tokens against 22,500), and recovers more gold figures: 15 of 16 in every run, against 10 to 13 for the store, whose three runs differ only on one question. There is no break-even against the vector index. Against reading the full corpus with the same model, the store's extraction cost is recovered after about 33 questions.

**Citation breadth is not citation correctness.** On the four contested questions a store answer cites 7.5 distinct papers and a vector-index answer 2.6, and neither arm cites a paper outside the corpus. Whether the additional citations support the statements they are attached to was not assessed on this fixture; §3.1 measures attribution on a benchmark where source records can be checked.

## 3.3 Derivation and withdrawal

Rules chained over stored assertions return each answer with the rule and the path that produced it; the paths trace graph edges, not the documents behind them. On PoLo's compound-treats-disease split of Hetionet (n=151, filtered ranking over all entities), 80 rules mined from the training pairs and pooled by noisy-or reach MRR 0.469 after pruning to 40 rules chosen on the development set (0.446 unpruned), against 0.402 reported for PoLo on the same split. [`hetionet`]

Withdrawal is recorded but does not yet propagate end to end. In a corpus built around one retracted paper, its retraction notice, three papers restating the finding and one control, tracing a claim to the notice requires a link from DOI to stored source that the system computes and does not persist, and the trace stops there. Which later claims a retraction should affect, given independent support, is not yet defined.

### 3.4 Adjudication

The store holds what adjudicating conflicting evidence requires, a problem with a long literature of its own [4, 12, 13, 30], but stored evidence does not by itself adjudicate well. On 2,261 items from six conflict benchmarks (ClinVar with ClinGen, RAMDocs, WikiContradict, Google CONFLICTS, MAGIC, and SemMedDB pairs), scored as binary item-weighted accuracy with contested items credited for disclosing the conflict, a router that discards hedged sides and ranks the rest by corroboration, source reliability and recency scores 0.722. That is above the best fixed response (0.354), `claude-opus-5` (0.693) and `gemini-2.5-flash` (0.570), and below a one-line rule at 0.768: drop hedged sides, answer if one definite side remains, otherwise report a conflict.

Both scores are exploratory. No split was held out, the router's priors were set by hand, and the rule was found after inspecting the data. Neither handles real disagreement: the rule scores 0.000 on the 509 resolvable items with two or more competing definite sides, while it scores 0.964 on the 1,801 items of the ten benchmark cells without them. Weighing competing definite evidence remains open.

### 3.5 Where the store loses: labelled disagreement

§3.1 scores attribution over predications an automatic extractor called opposed. A second benchmark asks the harder question, over disagreement that people found. ManConCorpus [1] takes 24 questions from cardiovascular systematic reviews and 254 PubMed abstracts, and its annotators marked, for each abstract, whether it answers its question yes or no. Every question has abstracts on both sides. We put the same 254 abstracts behind every arm, asked the same reader (Gemini 3.1 Flash-Lite, temperature 0) which papers answer yes and which answer no, and scored the reply against the annotators' identifiers, with no model judge. [`mancon`]

On the 16 held-out questions of the 24, the reader named the smaller side's papers and placed them as their annotator had at 0.65 and 0.91 from BM25's top twenty abstracts (8,988 tokens), 0.67 and 0.89 from the nearest twenty by embedding, and 0.65 and 0.92 from all 254 abstracts (113,709 tokens), against **0.40 and 0.78 from the store's claims** (14,805 tokens).

**Retrieving documents beats the store here, at half the prompt.** Reading every abstract scores no better than reading twenty, so 0.65 is what this reader achieves with nothing hidden from it. Three engine changes, made against a split of these questions fixed in advance, took the store from 0.30 to 0.40 and the papers reaching the reader from 0.49 to 0.72 while placing more of them wrongly (0.90 to 0.78): spreading the answer's budget across documents, scaling the bonus for a named entity by how specific it is, and recording each document's own conclusion as a claim.

The residue is a property of the representation. A conclusion sentence usually asserts two things and the claim recorded is often the incidental one; and these questions are comparisons — an intervention, against a comparator, on an outcome — while a triple has nowhere to put the comparator, which survives only in the quoted sentence.

**Table 3: Two runs per stage with a fixed model and settings. Live rows exclude four papers that the two builds extracted on different engine trees.**

| stage | corpus | two runs give |
|---|---|---|
| ingest, replayed | 8 papers, frozen extraction | 2,483 claims, 1,949 entities, agreement 1.0000 |
| extraction | 46 papers, live | 0 of 46 papers with the same emitted fact set; counts differ by 15.8% on average |
| finished store | 46 papers, live | 1,947 propositions shared of 6,261 and 6,272; Jaccard 0.18 |

## 4 REBUILD STABILITY

### 4.1 Three ways to reproduce a store

A store held in common can be reproduced in three ways: copied and verified, rebuilt by replaying recorded extractions, or re-derived from its documents by fresh model calls. Reproducibility, provenance and verifiability are recognized as jointly foundational to multi-party data management [34]; for a claim store the three operations differ. A copy needs integrity checks. Replay needs the recorded outputs. Re-derivation, which a second party needs to check a record against its sources, needs the extractor to decide the same way twice, and a language model does not.

### 4.2 Re-deriving a store from its documents

Table 3 contrasts replay with re-derivation.

Replaying a frozen extraction reproduces the store exactly, which places the non-determinism in extraction. Two builds of one 50-paper corpus with `gpt-5.6-luna` (temperature 1, fixed by the provider) and no identity resolution differ in every document. Both builds were stopped partway and resumed on a rebased engine tree, so four papers were extracted by the two builds on different trees and the resume re-ingested 1,320 and 1,572 duplicate claims; the figures here exclude those papers and count distinct propositions. On the 30 papers that both builds extracted on one commit, Jaccard is 0.21, but those are also the 30 longest papers, so the difference cannot be credited to the shared tree. Most of the difference is how a finding is written rather than what it says: on those 30 papers, 82% of one build's claims have a claim from the same source passage in the other, the paired claims read mostly state one finding with a different predicate or entity spelling, none states the opposite direction, and the builds agree on whether a result is null in 98% of same-passage pairs. What differs is coverage (16% of claims come from passages the other build skipped), one-sided errors concentrated in tables, and qualifiers. Asked eight questions, the two stores reached the same conclusions but not the same supporting figures (13 against 10 and 11 of 16), because a finding stored under another entity spelling is not joined without identity resolution. [`rebuild_divergence`] Comparable instability has been reported across seeds, languages and models [16] and across models extracting from one biomedical corpus [46]; here model, temperature and corpus are fixed.

**Table 4: Presence in the second build by number of source papers in the first, 46 papers. 95% Wilson intervals: 28.6–30.9, 73.7–88.8, 83.8–97.9. Per source: share of a proposition's source papers that re-extract it.**

| sources | first build | also in second | agreement | per source |
|---|---|---|---|---|
| one | 6,114 | 1,820 | 29.8% | 29.1% |
| two | 97 | 80 | 82.5% | 57.7% |
| three+ | 50 | 47 | 94.0% | 68.5% |

**Table 5: Stores built from the same documents in two opposite ingestion orders, one run each. Agreement: Jaccard over the multiset of content-derived keys. Keys differing: keys present in only one store.**

| documents | claims, A / B | agreement | keys differing |
|---|---|---|---|
| 8 | 2,483 / 2,490 | 0.9860 | 35 (1.4%) |
| 16 | 4,498 / 4,514 | 0.9746 | 116 (2.5%) |

**Agreement is higher for propositions asserted by more papers** (Table 4).

The association decomposes into two parts. Each source is a chance to re-extract the proposition, and the per-source rate is itself higher for propositions with more sources. Treating sources as independent at each bucket's own rate predicts 82% and 98% agreement, close to what was observed; at the single-source rate it predicts 50% and 73%. Corroborated propositions are therefore both easier to re-extract and extracted more often, and these data do not separate corroboration from salience. Source counts are unstable too: 40 single-source propositions reappear only through a different paper. With 97.7% of propositions asserted once and two builds in all, the intervals cover propositions within one pair of builds, not variation between builds. [`rebuild_clean_subset`, `corroboration_vs_rebuild`]

## 4.3 Concurrent writers and ingestion order

[`two_writers.json`] Writer and order effects were measured by replaying frozen extractions, so that the runs differ only in writer and in order.

**The engine excludes a second writer.** Of two processes ingesting disjoint document sets into one store, the second was refused the write lock when it opened the store and wrote nothing; the store held exactly the first writer's 1,381 claims. This is safe exclusion, not concurrent contribution: one party's documents were not added.

**The store depends on ingestion order** (Table 5). The same documents ingested in two opposite orders, once each, yield different stores.

The differences are in content rather than generated identifiers: about half are meta-claims, and the base claims that differ pair up as surface forms of one entity (`tnf` and `tumor necrosis factor-alpha`). Two runs in the same order, at eight documents only, differed on no key. The mechanism is that of §2.3: the first document to introduce a surface form establishes the entity that later documents resolve against. The differing share was larger at sixteen documents than at eight; two sizes and one permutation each do not establish how it grows.

## 4.4 Replay is not re-derivation

A shared artifact need not be rebuilt from its sources to be useful: a copied store can be verified and queried as it stands, and replaying frozen extractions reproduced the finished store exactly (Table 3). Re-deriving a store from its documents is a different operation, and it is the one that fails. Where independent re-extraction is required, as when a second party checks the record against the sources, lowering temperature or voting samples an undecided model again rather than fixing a decision. The alternative is to retain the model's write-time decisions and replay them. That reproduces decisions, not their correctness: a retained wrong decision replays perfectly. A retained decision is therefore useful only if it is written where it can be reviewed and contested, beside the corpus and versioned with it, which the term readings of §2.3 do for one class of decision.

## 4.5 Limits of these results

**The builds are one pair on one corpus.** Agreement by source count is an association observed across two builds; its intervals reflect sampling of propositions, not variation between builds, and the top bucket is small.

**The 50-paper builds did not run identity resolution** and were resumed on a different engine tree partway through (§4.2). They characterize extraction under that configuration, not the complete write path of §2.

**Several stability observations are reported but not reproduced here.** Structure assignment gave five distinct proposals across runs, temperature 0 and voting did not change that, and replaying recorded structure decisions reproduced nine of nine tables; an earlier 14-paper store showed 31% of its merge set changing across identical identity runs. These come from project records without committed raw outputs and are not used as evidence above.

**The concurrency experiment is small**: at most sixteen replayed documents, one permutation per size.

A pre-registered experiment addresses the first two limits: one nested corpus at three sizes, each built twice, plus an arm that replays the first build's recorded decisions. Predictions and grading rules were committed on 2026-09-11, before any build; the 50-paper builds reported here are its pilot, and their deviations are recorded there. [`PRE_REGISTRATION.md`]

# 5 RESEARCH AGENDA

Sections 2 to 4 describe a record that one party builds and many read. Shared use, with several people or agents contributing to one fact set, requires three capabilities the system does not yet provide.

**Portable, reviewable decision records.** Section 4 shows that replaying frozen extractions reproduces a store while re-deriving it from documents does not, and proposes retaining a model's write-time decisions so they can be replayed. Those records have to travel with the corpus, since a second deployment decides again and diverges, and to be reviewable, since replaying a judgement makes it stable rather than correct. A legible, versioned record distributed with the corpus, as the documentation literature argues for models

and datasets [35], satisfies both; a hash-keyed cache satisfies only the first.

**Identity that stays correct and cheap as the store grows.** The cost of identity decisions scales with the store rather than with the incoming document. Settling orthographic variants without a model and deciding identity per term rather than per mention should remove most model calls, but neither has been measured for merge precision and recall, which is what decides whether removing calls is safe.

**A model of concurrent contribution.** The engine admits one writer and refuses a second (§4.3), so multi-party use requires taking turns, and the order of turns changed the resulting store. Two questions precede reconciliation between organizations: what the unit of concurrent contribution is, and whether identity can be decided without privileging the first writer. A store whose content depends on ingestion order is shared only by convention.

## 6 CONCLUSION

Storing one row per source assertion, keyed by proposition, turns counting sources and finding potentially contested evidence into group-bys. On opposed biomedical predications too numerous to read whole, most of which a sampled reading found to be extraction errors rather than disagreements, answers built from the store's records cited no source from outside the question, and ablations show that most of the benefit comes from indexing each source's assertion by concept pair, which a relational schema can also provide; the remainder depends on a declared opposition vocabulary that the benchmark shares.

What it does not buy is a better reading of the documents themselves: asked which papers concluded yes and which no, a reader given twenty retrieved abstracts beat the same reader given the store's claims, at half the prompt.

A record two parties share must also be reproducible, and which operation is meant matters. Replaying frozen extractions reproduced a store exactly; re-deriving it from its documents did not, with two builds sharing under a fifth of their propositions. Corroborated propositions reappeared more often, largely because they were easier to re-extract and had more chances to be. Retaining and reviewing write-time decisions, rather than re-deriving them, is the agenda of §5.

## REFERENCES


[1] Abdulaziz Alamri and Mark Stevenson. 2016. A corpus of potentially contradictory research claims from cardiovascular research abstracts. *Journal of Biomedical Semantics* 7, 36 (2016). https://doi.org/10.1186/s13326-016-0083-z

[2] Renzo Angles. 2018. The Property Graph Database Model. In *Proceedings of the 12th Alberto Mendelzon International Workshop on Foundations of Data Management (AMW) (CEUR Workshop Proceedings)*, Vol. 2100. https://ceur-ws.org/Vol-2100/paper26.pdf

[3] Berk Atil, Sarp Aykent, Alexa Chittams, Lisheng Fu, Rebecca J. Passonneau, Evan Radcliffe, Guru Rajan Rajagopal, Adam Sloan, Tomasz Tudrej, Ferhan Ture, Zhe Wu, Lixinyu Xu, and Breck Baldwin. 2024. Non-Determinism of "Deterministic" LLM Settings. *arXiv preprint arXiv:2408.04667* (2024).

[4] Jens Bleiholder and Felix Naumann. 2009. Data fusion. *Comput. Surveys* 41 (2009), 1–41. https://doi.org/10.1145/1456650.1456651

[5] Peter Buneman, Sanjeev Khanna, and Wang-Chiew Tan. 2001. Why and Where: A Characterization of Data Provenance. In *Proceedings of the 8th International Conference on Database Theory (ICDT)*. 316–330. https://doi.org/10.1007/3-540-44503-x_20

[6] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam Hruschka, and Tom Mitchell. 2010. Toward an Architecture for Never-Ending Language Learning. *Proceedings of the AAAI Conference on Artificial Intelligence* 24 (2010), 1306–1313. https://doi.org/10.1609/aaai.v24i1.7519

[7] Jeremy J. Carroll, Christian Bizer, Pat Hayes, and Patrick Stickler. 2005. Named graphs, provenance and trust. In *Proceedings of the 14th International Conference on World Wide Web (WWW)*. 613. https://doi.org/10.1145/1060745.1060835

[8] James Cheney, Laura Chiticariu, and Wang-Chiew Tan. 2009. Provenance in Databases: Why, How, and Where. *Foundations and Trends in Databases* 1 (2009), 379–474. https://doi.org/10.1561/9781601982339

[9] Tim Clark, Paolo N. Ciccarese, and Carole A. Goble. 2014. Micropublications: a semantic model for claims, evidence, arguments and annotations in biomedical communications. *Journal of Biomedical Semantics* 5, 1 (2014), 28. https://doi.org/10.1186/2041-1480-5-28

[10] Richard Cyganiak, David Wood, and Markus Lanthaler. 2014. RDF 1.1 Concepts and Abstract Syntax. W3C Recommendation. https://www.w3.org/TR/rdf11-concepts/

[11] Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 2014. Knowledge vault. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 601–610. https://doi.org/10.1145/2623330.2623623

[12] Xin Luna Dong, Laure Berti-Equille, and Divesh Srivastava. 2009. Integrating conflicting data. *Proceedings of the VLDB Endowment* 2 (2009), 550–561. https://doi.org/10.14778/1687627.1687690

[13] Xin Luna Dong and Felix Naumann. 2009. Data fusion. *Proceedings of the VLDB Endowment* 2 (2009), 1654–1655. https://doi.org/10.14778/1687553.1687620

[14] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2023. Retrieval-Augmented Generation for Large Language Models: A Survey. *arXiv preprint arXiv:2312.10997* (2023).

[15] Lise Getoor and Ashwin Machanavajjhala. 2012. Entity resolution. *Proceedings of the VLDB Endowment* 5 (2012), 2018–2019. https://doi.org/10.14778/2367502.2367564

[16] Luca Giordano and Simon Razniewski. 2026. Foundations of LLM Knowledge Materialization: Termination, Reproducibility, Robustness. In *Findings of the Association for Computational Linguistics: EACL 2026*.

[17] Todd J. Green, Grigoris Karvounarakis, and Val Tannen. 2007. Provenance semirings. In *Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. 31–40. https://doi.org/10.1145/1265530.1265535

[18] Alon Halevy, Flip Korn, Natalya F. Noy, Christopher Olston, Neoklis Polyzotis, Sudip Roy, and Steven Euijong Whang. 2016. Goods. In *Proceedings of the 2016 International Conference on Management of Data*. 795–806. https://doi.org/10.1145/2882903.2903730

[19] Olaf Hartig, Pierre-Antoine Champin, and Andy Seaborne. 2026. RDF 1.2 Concepts and Abstract Data Model. W3C Candidate Recommendation Snapshot. https://www.w3.org/TR/rdf12-concepts/ Snapshot of 7 April 2026. Accessed 2026-09-14.

[20] Melanie Herschel, Ralf Diestelkämper, and Houssem Ben Lahmar. 2017. A survey on provenance: What for? What form? What from? *The VLDB Journal* 26 (2017), 881–906. https://doi.org/10.1007/s00778-017-0486-1

[21] Yujia Hu, Tuan-Phong Nguyen, Shrestha Ghosh, and Simon Razniewski. 2025. Enabling LLM Knowledge Analysis via Extensive Materialization. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*.

[22] Silu Huang, Liqi Xu, Jialin Liu, Aaron J. Elmore, and Aditya Parameswaran. 2017. O rpheus DB. *Proceedings of the VLDB Endowment* 10 (2017), 1130–1141. https://doi.org/10.14778/3115404.3115417

[23] Moe Kayali, Anton Lykov, Ilias Fountalis, Nikolaos Vasiloglou, Dan Olteanu, and Dan Suciu. 2024. Chorus: Foundation Models for Unified Data Discovery and Exploration. *Proceedings of the VLDB Endowment* 17 (2024), 2104–2114. https://doi.org/10.14778/3659437.3659461

[24] Halil Kilicoglu, Dongwook Shin, Marcelo Fiszman, Graciela Rosemblat, and Thomas C. Rindflesch. 2012. SemMedDB: a PubMed-scale repository of biomedical semantic predications. *Bioinformatics* 28, 23 (2012), 3158–3160. https://doi.org/10.1093/bioinformatics/bts591

[25] Tobias Kuhn and Michel Dumontier. 2014. Trusty URIs: Verifiable, Immutable, and Permanent Digital Artifacts for Linked Data. In *The Semantic Web: Trends and Challenges (ESWC 2014) (Lecture Notes in Computer Science)*. 395–410. https://doi.org/10.1007/978-3-319-07443-6_27

[26] Tobias Kuhn, Ruben Taelman, Vincent Emonet, Haris Antonatos, Stian Soiland-Reyes, and Michel Dumontier. 2021. Semantic micro-contributions with decentralized nanopublication services. *PeerJ Computer Science* 7 (2021), e387. https://doi.org/10.7717/peerj-cs.387

[27] Krishna Kulkarni and Jan-Eike Michels. 2012. Temporal features in SQL:2011. *ACM SIGMOD Record* 41, 3 (2012), 34–43. https://doi.org/10.1145/2380776.2380786

[28] Timothy Lebo, Satya Sahoo, and Deborah McGuinness. 2013. PROV-O: The PROV Ontology. W3C Recommendation. http://www.w3.org/TR/2013/REC-prov-o-20130430/

[29] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*.

[30] Yaliang Li, Jing Gao, Chuishi Meng, Qi Li, Lu Su, Bo Zhao, Wei Fan, and Jiawei Han. 2016. A Survey on Truth Discovery. *ACM SIGKDD Explorations Newsletter* 17 (2016), 1–16. https://doi.org/10.1145/2897350.2897352

[31] Yuliang Li, Jinfeng Li, Yoshihiko Suhara, AnHai Doan, and Wang-Chiew Tan. 2020. Deep entity matching with pre-trained language models. *Proceedings of the VLDB Endowment* 14 (2020), 50–60. https://doi.org/10.14778/3421424.3421431

[32] Michael Maddox, David Goehring, Aaron J. Elmore, Samuel Madden, Aditya Parameswaran, and Amol Deshpande. 2016. Decibel. *Proceedings of the VLDB Endowment* 9 (2016), 624–635. https://doi.org/10.14778/2947618.2947619

[33] Laura Menotti, Stefano Marchesin, Fabio Giachelle, and Gianmaria Silvello. 2025. Provenance-driven nanopublications: representing source lineage and trust networks for multi-source assertions. *International Journal on Digital Libraries* 26, 4 (2025). https://doi.org/10.1007/s00799-025-00431-x Article 24.

[34] Sergii Mikhtoniuk and Ozge Nilay Yalcin. 2021. Open Data Fabric: A Decentralized Data Exchange and Transformation Protocol With Complete Reproducibility and Provenance. *arXiv preprint arXiv:2111.06364* (2021).

[35] Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. 2019. Model Cards for Model Reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*. 220–229. https://doi.org/10.1145/3287560.3287596

[36] T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, B. Yang, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, and J. Welling. 2018. Never-ending learning. *Commun. ACM* 61 (2018), 103–115. https://doi.org/10.1145/3191513

[37] Sidharth Mudgal, Han Li, Theodoros Rekatsinas, AnHai Doan, Youngchoon Park, Ganesh Krishnan, Rohit Deep, Esteban Arcaute, and Vijay Raghavendra. 2018. Deep Learning for Entity Matching. In *Proceedings of the 2018 International Conference on Management of Data*. 19–34. https://doi.org/10.1145/3183713.3196926

[38] Avanika Narayan, Ines Chami, Laurel Orr, and Christopher Ré. 2022. Can Foundation Models Wrangle Your Data? *Proceedings of the VLDB Endowment* 16 (2022), 738–746. https://doi.org/10.14778/3574245.3574258

[39] Vinh Nguyen, Olivier Bodenreider, and Amit Sheth. 2014. Don't Like RDF Reification? Making Statements about Statements Using Singleton Property. In *Proceedings of the 23rd International Conference on World Wide Web (WWW)*. 759–770. https://doi.org/10.1145/2566486.2567973

[40] George Papadakis, Dimitrios Skoutas, Emmanouil Thanos, and Themis Palpanas. 2020. Blocking and Filtering Techniques for Entity Resolution. *Comput. Surveys* 53 (2020), 1–42. https://doi.org/10.1145/3377455

[41] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language Models as Knowledge Bases?. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 2463–2473. https://doi.org/10.18653/v1/d19-1250

[42] Alexander Ratner, Stephen H. Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. 2017. Snorkel. *Proceedings of the VLDB Endowment* 11 (2017), 269–282. https://doi.org/10.14778/3157794.3157797

[43] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago. In *Proceedings of the 16th international conference on World Wide Web*. 697–706. https://doi.org/10.1145/1242572.1242667

[44] Denny Vrandečić and Markus Krötzsch. 2014. Wikidata. *Commun. ACM* 57 (2014), 78–85. https://doi.org/10.1145/2629489

[45] Wikidata contributors. [n.d.]. Help:QuickStatements. https://www.wikidata.org/wiki/Help:QuickStatements Accessed 2026-09-14.

[46] Qiang Xie, Yang Fang, and Yanan Hu. 2026. LLMs build inconsistent knowledge graphs from one immune-checkpoint corpus: a five-axis benchmark and consensus precision–coverage protocol. *Frontiers in Immunology* (2026).