HAVE WE SEEN THIS BEFORE? EVALUATING AND IMPROVING INCREMENTAL KNOWLEDGE GRAPH CONSTRUCTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Knowledge graphs organize information from multiple sources for querying and reasoning. As new text arrives, maintaining a coherent graph requires integrating equivalent information without duplicating it and preserving distinct information without incorrectly merging it. Evaluating these complementary requirements is challenging because the appropriate update depends on what the graph already contains. We introduce DéjàKG, a controlled benchmark spanning initial Construction, Consolidation of equivalent information, and Extension with distinct new information. Single-fact inputs and predefined identity relationships specify expected graph updates, enabling deterministic, case-wise scoring without an additional large language model (LLM) judge. To address these integration challenges, we introduce IDEM, a framework that determines whether incoming entities and facts match existing knowledge or should be added to the graph. It compares their surface forms and descriptions using vector similarity, LLM-based decisions, or a hybrid of the two.

1 INTRODUCTION

Knowledge graphs (KGs) bring information from multiple sources into a shared structure for querying and reasoning (Hogan et al., 2021). Their usefulness depends not only on the facts extracted from each source, but also on how those facts are connected. In incremental construction, new text must therefore be interpreted in relation to the entities and facts already represented in the graph.

An organization's full name and acronym may denote the same entity; creating separate nodes for them fragments its information (Vashishth et al., 2018). Conversely, "Mercury" may denote a planet or an element, and merging the two conflates unrelated information. The same distinction matters for facts: a paraphrase can repeat an existing assertion, whereas a change in a relation or value can introduce a different assertion. Integration thus requires more than recognizing similar wording.

Existing evaluations examine extraction quality (Mihindukulasooriya et al., 2023) and the resolution of entity and relation representations (Jiang et al., 2023; Lairgi et al., 2025). We ask a complementary question: does incoming text lead to appropriate reuse of existing entities and facts or addition of distinct information? This judgment depends on the graph already built. If an organization was never added under its full name, adding it later under its acronym is not a missed merge. An evaluation must therefore distinguish missing initial information from failures of subsequent integration.

We introduce DéjàKG, a controlled benchmark with an accompanying evaluation protocol and Valid Update (VU) score. Construction tests initial graph building. Consolidation tests reuse of existing entities or facts, including new facts about known entities. Extension tests preservation of distinct information despite potentially misleading similarities. Single-fact inputs with controlled reuse and novelty specify the expected numbers of new entities and facts for each input. Comparing these with observed additions enables deterministic, case-wise scoring without an additional LLM judge; VU summarizes performance across the three tasks. The protocol accommodates different output wording under shared entity/fact granularity. Counts remain structural diagnostics, not verification of semantic correctness (Section 4).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

To address the same integration problem, we introduce IDEM, an incremental KG construction framework. It generates descriptions that identify entities and express relation meaning, direction, and argument roles. These descriptions complement names when retrieving and comparing possible matches, providing information beyond surface wording. Integration can use vector similarity, LLM-based decisions, or a hybrid policy that combines selective vector-based fact matching with LLM fallback. When an LLM is used for integration, it receives retrieved candidates rather than the entire graph. Held-out results show higher count-based update scores for IDEM's LLM-based and hybrid policies than for its vector policy.

2 RELATED WORK

From extraction to incremental construction. Text-to-KG construction combines information extraction with integration across sources (Hogan et al., 2021). OpenIE extracts relational tuples without a predefined relation inventory (Angeli et al., 2015); REBEL formulates triple extraction as sequence-to-sequence generation (Huguet Cabot & Navigli, 2021). Beyond extraction, iText2KG integrates incoming information into an existing graph (Lairgi et al., 2025). IDEM addresses this integration setting (Section 5).

Canonicalization and identity resolution. Canonicalization reconciles alternative representations of entities and relations. CESI and KGGen use clustering for this purpose (Vashishth et al., 2018; Mo et al., 2025). EDC generates contextual definitions and combines embedding-based retrieval with LLM verification to canonicalize schema elements (Zhang & Soh, 2024).

Integrating incoming information involves two decisions: whether to reuse an existing entity and whether a complete fact is already represented. iText2KG uses embedding-based entity and relation matching (Lairgi et al., 2025). Like EDC, IDEM uses generated descriptions; here they support entity resolution and complete-fact comparison during incremental updates, with separate name and description embeddings for matching.

Evaluating extraction and integration. Evaluation likewise depends on the unit being assessed. At the extraction level, CaRB scores tuples, and Text2KGBench evaluates ontology-guided extraction, conformance, and hallucination (Bhardwaj et al., 2019; Mihindukulasooriya et al., 2023). For canonicalization and resolution, COMBO evaluates noun and relation phrase canonicalization against annotated clusters (Jiang et al., 2023), and iText2KG reports resolution errors (Lairgi et al., 2025). Related entity-linking work addresses previously unknown entities: EDIN tests their discovery and indexing for subsequent linking (Kassner et al., 2022). At the graph level, KGGen's MINE evaluates information recovery from generated graphs (Mo et al., 2025).

DéjàKG complements these evaluations with controlled tests of individual graph updates. Like CheckList's capability-oriented tests (Ribeiro et al., 2020), its categories target specific behaviors: reusing equivalent information and preserving distinct information despite confusable wording (Section 3). Rather than matching outputs to gold tuples or clusters, the protocol scores entity and fact additions relative to each system's own graph (Section 4). This comparison requires shared entity/fact granularity; matching counts alone does not establish semantic correctness.

3 THE DÉJÀKG BENCHMARK

3.1 CONTROLLED GRAPH-UPDATE TASKS

DéjàKG tests initial construction, reuse of equivalent information, and preservation of confusable new information. Each input expresses one subject–predicate–object assertion, called a *fact* or *claim* without certifying its truth. For each input, controlled reuse and novelty specify the expected numbers of new entities and facts (Section 4). Systems receive only the input text.

Construction starts from an empty graph and is designed to add two distinct referents and one fact per input, with no referent shared across cases. With N cases, the intended graph contains $2N$ referents. This enlarges the pool of possible matches, not necessarily the difficulty. Sparse connectivity is deliberate: fragmentation here means duplicate representations of one identity, not disconnected components.

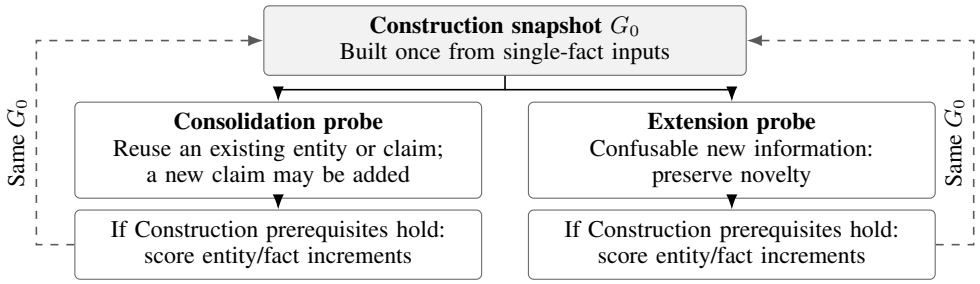

Figure 1: Independent probe evaluation. Each system first builds its Construction graph G_0. Every probe starts from that same graph; its changes are not passed to later probes. Probes are scored when their required Construction information passes the protocol's checks, even if probe extraction fails. These checks do not verify semantic correctness.

A *probe* is a test input linked to designated Construction knowledge, its *anchor*. Consolidation tests both reuse of an existing fact and reuse of an existing entity in a new fact. For example, if the graph already contains the World Health Organization but not Geneva, "WHO is headquartered in Geneva" should reuse the organization node while adding Geneva and the new fact.

Extension tests whether a distinct new fact is preserved, even when both entities already exist. For example, if the graph already contains "A supports B," the input "A opposes B" should add a distinct fact while reusing the existing nodes for A and B.

Let G_0 denote the graph produced by each system after processing all Construction inputs. Each probe is evaluated independently against this fixed snapshot (Figure 1). We measure the entity and fact increments produced by the system's update procedure, without carrying probe-induced changes forward to subsequent probes. The design retains intended probe–Construction links and aims to avoid unintended sharing. The protocol tests separate updates, not cumulative streams or temporal truth revision; Section 4 defines scoring.

3.2 Category design and rationale

Following capability-oriented testing (Ribeiro et al., 2020), we organize ten categories around entity names, relation meaning and direction, and the objects or values stated in a claim. The categories may overlap; they neither cover all possible cases nor represent their frequency in natural text. We summarize their design below; Appendix A gives authoring details and Table 3 pairs each category's purpose with complete input examples.

Names versus referents. Naming variation motivates KG canonicalization (Vashishth et al., 2018). Acronym, alias, and multilingual probes change how an existing entity is named and test whether it is reused. Homonym and near-entity probes instead introduce different entities with identical or similar names. Together, they test whether a system distinguishes name similarity from entity identity.

Relation meaning and direction. Relation-synonym probes express the same claim in different words. Inverse probes also preserve the claim, but reverse the arguments and use an inverse predicate, as in "A is B's parent" and "B is A's child." Comparing only the original direction can miss this equivalence. By contrast, antonym probes keep the argument order but replace the predicate with its opposite, changing the claim. Antonyms can occur in similar contexts and remain close in static word embeddings (Nguyen et al., 2016); these probes test whether a system distinguishes opposition from equivalence despite that similarity.

Additional objects and changed values. Sibling probes add another object for the same subject and predicate, such as another coauthor of a book; both facts can hold simultaneously. Here, sibling does not refer to a taxonomic relationship. Different-value probes instead substitute a value, such as changing a capacity from 8 GB to 16 GB. They test preservation of the distinct assertion, not which value is true. In both cases, shared wording should not cause the new fact to be merged with the

existing one. These distinctions motivate IDEM's entity and relation descriptions (Section 5), but category scores alone do not establish the causes of success or failure.

3.3 CANDIDATE SELECTION, GENERATION, AND CURATION

We used Wikidata and WordNet as sources of candidate seeds (Vrandečić & Krötzsch, 2014; Miller, 1995), supplemented by source-derived facts and author-added candidates. We also include authored fictional entities to test integration beyond familiar real-world facts. Before sentence generation, the author screened candidates against category-specific criteria; inclusion in a source resource alone does not establish a valid probe. These seeds differ from the Construction anchors used in evaluation.

Selected seeds guide AI-assisted drafts under author-specified generation rules. Probes can also be formed by transforming Construction inputs, including inverse wording, multilingual names, and value substitution. Construction sentences also incorporate deterministic, relation-specific rewrites of source-derived facts. Sentence construction thus combines generation with controlled rewriting.

The author iteratively reviewed and revised generated sentences with AI assistance against category-specific criteria. AI agents flagged potential errors and assisted with revisions, while the author made the final decisions on acceptance, correction, or rejection. Appendix A.1 describes the generation and review procedures in detail.

Dataset size. The development split contains 5,000 Construction, 500 Consolidation, and 500 Extension inputs; the test split contains 1,000, 250, and 250, respectively, for 7,500 inputs in total. Within each split, Consolidation and Extension are each balanced across their five categories, with 100 inputs per category in development and 50 in test.

4 EVALUATION PROTOCOL

For each single-fact input, we compare the numbers of entities and facts added to the graph with the expected counts. The expected counts depend on which entities and facts should already be present. Evaluating inputs separately prevents excess additions on one input from compensating for missing additions on another. Restricting inputs to a single fact keeps updates for different claims from being combined in the same count change.

Because scoring uses count changes rather than matching outputs to gold triples, it can accommodate differences in how systems express the same information. For example, "The book was written by Alice" may yield `(the book, written_by, Alice)` or `(Alice, wrote, the book)`. Within each comparison condition, each system uses a fixed extraction configuration across Construction, Consolidation, and Extension. We measure updates relative to each system's own graph, while requiring the same entity/fact granularity across systems.

This count-based scoring requires no additional LLM judge and is deterministic for fixed system outputs. It therefore avoids dependence on a judge's accuracy or response variability. However, incorrect entities or links can still receive credit if the counts match the expected additions.

Let D_0, D_C, D_X denote the Construction, Consolidation, and Extension cases. For input i, ΔV_i and ΔF_i are the observed numbers of entities and facts newly added to the graph, respectively. Construction updates are measured sequentially; each probe is measured independently against the same Construction snapshot (Section 3).

4.1 CONSTRUCTION AND CONDITIONAL PROBE SUCCESS

All three tasks use the same extraction check: $Q_i = 1$ if the system extracts exactly one fact from input i and records a fact-level output, and $Q_i = 0$ otherwise. This counts extracted facts, not facts newly added to the graph: an extracted fact can match an existing fact and be reused. Success additionally requires the numbers of newly added entities and facts to match their expected values. Appendix B.1 specifies the extraction check. We use $\mathbf{1}[\cdot]$ for the indicator function.

An entity-disjoint Construction case should add two entities and one fact. Its success indicator a_i and the Construction score C are

$$a_i = Q_i \, \mathbf{1}[\Delta V_i = 2 \wedge \Delta F_i = 1], \qquad C = \frac{1}{|D_0|} \sum_{i \in D_0} a_i. \tag{1}$$

An incorrect merge can also reduce C, since Construction is sequential; the score does not measure extraction alone.

For Consolidation and Extension, the intended reuse or distinction can be tested only when the required prior information is available. We use the Construction checks as a proxy for this prerequisite: $B_i = 1$ when all required count and extraction checks pass, and $B_i = 0$ otherwise. These checks cover the designated anchor components and any other entities or facts the probe should reuse. Entity and fact requirements are checked separately: a probe requiring only entities can be eligible even if its originating Construction case failed the fact-count check. These checks determine eligibility, not semantic correctness of the required information (Appendix B.1).

For an eligible probe, success requires the extraction check and both expected increments to match. With case-specific expected increments (v_i^*, f_i^*), define

$$S_i = Q_i \, \mathbf{1}[\Delta V_i = v_i^* \wedge \Delta F_i = f_i^*]. \tag{2}$$

The expected increments can be $(0, 0)$ when the entire claim should be reused. For an eligible probe, extracting zero or multiple facts is a failure to process the input, not a reason to exclude it from evaluation.

The Entity and Claim Consolidation Rate (ECR) and Entity and Claim Extension Rate (EER) are the fractions of eligible probes that succeed:

$$\mathrm{ECR} = \frac{\sum_{i \in D_C} B_i S_i}{\sum_{i \in D_C} B_i}, \qquad \mathrm{EER} = \frac{\sum_{i \in D_X} B_i S_i}{\sum_{i \in D_X} B_i}. \tag{3}$$

ECR and EER therefore measure count-based success from input text to graph update, not integration decisions in isolation.

Scores pool cases rather than average category rates; an empty eligible set yields N/A. We report successful, eligible, and total case counts alongside the scores (Section 6.2).

4.2 Valid Update score and interpretation

The three tasks assess complementary capabilities: initial construction, reuse of equivalent information, and preservation of distinct information. We assign them equal importance and combine their scores using a harmonic mean, which emphasizes weaker components. For positive component scores, the Valid Update (VU) score is

$$\mathrm{VU} = \frac{3}{C^{-1} + \mathrm{ECR}^{-1} + \mathrm{EER}^{-1}}. \tag{4}$$

If all three component scores are defined and at least one is zero, we report $\mathrm{VU} = 0$. If any component is unavailable, for example because no probes are eligible, we report $\mathrm{VU} = \text{N/A}$ instead.

VU summarizes performance across the three tasks; it does not estimate the probability of succeeding at all three tasks.

Here, *valid* means passing the protocol's checks, not establishing semantic correctness. Even when the extraction check passes, a system could omit a required entity and add an unrelated one while still producing the expected entity increment. Section 6.4 examines selected extraction and integration errors, including an incorrect relation hidden by matching counts.

5 IDEM: Identity-Oriented Incremental Integration

IDEM first extracts information from text, then integrates it into the current graph (Figure 2). Generated entity and relation descriptions support both candidate retrieval and identity decisions. The vector, LLM-based, and hybrid variants share this extraction and representation pipeline, but differ in candidate selection and reuse-or-create decisions.

5.1 Triple Extraction and Identity-Oriented Descriptions

The default two-stage pipeline begins with an LLM extracting subject–predicate–object triples using instructions adapted from iText2KG (Lairgi et al., 2025). Equivalent claims can be harder to match when extracted in different directions. We supplement these instructions with direction guidance for property relations: the default prompt asks the model to place the entity being described first: "College owns the Hall" becomes `(Hall, owner, College)`. A second stage jointly generates entity and relation descriptions from the source and fixed triples, without changing their content, order, or direction. Neither stage reads the graph. The stages are sequential within a document, but extraction can run in parallel across documents before integration in stream order. Applications can override the default extraction conventions through entity/fact rules (Appendix B).

Entity names alone do not establish identity: the same name may denote different entities, while different names may denote the same entity. We ask the model to use source context to identify the intended referent and generate a definition that makes its identity explicit. The prompt favors stable identifying characteristics over a restatement of the current fact, aiming to keep descriptions comparable across different statements about the same entity. It also asks the model to preserve identity-bearing names and exact values rather than reduce distinct referents to generic types. When an adequate definition cannot be formed, the current relation and counterpart may supply additional identifying context. These instructions aim to support identity comparison; they do not guarantee consistent or correct descriptions.

For each extracted triple's fixed direction, relation descriptions aim to reduce wording variation while preserving meaning and argument roles. For example, `(Book, author, Person)` is described as `OBJECT is the author of SUBJECT`; describing only "authorship" would lose who wrote what. The prompt requests source-grounded templates with consistent wording for equivalent directed relations and distinct wording for opposing predicates. The `SUBJECT`/`OBJECT` placeholders remove concrete entity names without discarding their roles. These instructions guide description generation but do not guarantee correct downstream matches.

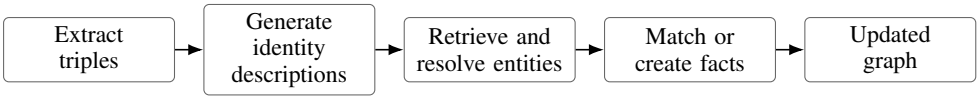

Figure 2: Conceptual IDEM pipeline for a document. Variants differ in candidate selection and reuse-or-create decisions (Section 5.3); gold identities and category labels are not system inputs.

5.2 Multi-view candidate retrieval

Descriptions provide identifying context and directed relation meaning, while names and predicate wording retain lexical cues that generated descriptions may omit. We embed the two views separately and weight their contributions to control the balance between lexical and descriptive evidence. For unit-normalized name and description vectors u_n, u_d and weight $0 \le w_e \le 1$,

$$\begin{aligned} z(e) &= [\sqrt{w_e}\, u_n(e); \sqrt{1-w_e}\, u_d(e)], \\ \mathrm{sim}(e, e') &= \langle z(e), z(e') \rangle = w_e \langle u_n(e), u_n(e') \rangle + (1-w_e) \langle u_d(e), u_d(e') \rangle. \end{aligned} \tag{5}$$

For nonzero unit-normalized views, cosine similarity equals the weighted sum above. Relations combine predicate wording and directed description analogously, with weight w_r. Unlike vector averaging, this construction introduces no cross-view similarity terms. Weights are empirical choices; descriptions do not guarantee separation of distinct identities. Entity retrieval uses Faiss's CPU `IndexFlatIP` for exhaustive inner-product search over all indexed entity vectors (Douze et al., 2026). Approximate indexing could reduce search time at the cost of missing high-scoring candidates, potentially altering integration decisions; this trade-off is not evaluated here.

5.3 Vector, LLM-based, and hybrid policies

Integration first resolves entities, then compares facts between the resolved endpoints. Fact comparison does not reconsider alternative entity matches. The three policies differ in how they retrieve candidates and decide whether to reuse an existing entity or fact.

Before insertion, the *vector policy* retrieves entity candidates for all document mentions against the pre-update graph. It reuses an entity when its global top-1 similarity meets the entity threshold. For facts, it searches only edges from the resolved subject to the resolved object and reuses the top-1 match if its relation similarity meets the relation threshold. Otherwise, it creates a new entity or fact. The directional restriction preserves argument roles: "A follows B" and "B follows A" are distinct claims despite identical predicate wording. Vector matching therefore depends on equivalent formulations being extracted with the same endpoint order; opposite-direction facts are not compared. Changing extraction instructions does not change this matching restriction. Distinct relations with the same ordered endpoints can still be falsely merged.

The *LLM-based policy* combines all entities registered under the mention's normalized name with the global semantic top-k. The LLM selects an existing entity or decides to create one; an empty candidate set yields a new entity directly. Equal spelling does not force reuse, and the combined set can exceed k. For facts, it ranks edges in both directions between the resolved endpoints and asks the LLM to compare the top-k. This permits comparison of inverse formulations, although retrieval can still omit the correct match.

The *identity-hybrid policy* follows LLM-based entity resolution, with automatic entity matching disabled by default. Automatic fact reuse has the same directional restriction as the vector policy; reverse-direction matches require LLM fallback. For the two highest relation similarities s_1, s_2 among edges from the resolved subject to the resolved object, reuse requires

$$\text{same ordered endpoints} \ \wedge \ s_1 \geq \tau_r \ \wedge \ s_1 - s_2 \geq m_r. \tag{6}$$

The threshold requires a close match; the margin requires separation from the runner-up. If the criterion is not met, the policy uses the LLM-based policy's comparison across both directions, rather than declaring the fact new. An empty candidate set yields a new fact. For a singleton candidate, $s_2 = -1$, so the margin does not test competition. Whether the margin helps under dense relation competition is not established by this benchmark.

In the LLM-based and hybrid policies, an endpoint resolved as new bypasses existing-fact retrieval and comparison: a genuinely novel endpoint cannot belong to an existing binary fact. An incorrect new-entity decision can therefore produce a duplicate fact. Remaining entity and fact decisions form separate LLM batches per document to limit sequential requests; automatically reused facts do not enter the fact batch. Reducing batch size avoids a call only when the batch becomes empty. Section 6.3 discusses representation and policy effects.

6 Experimental Evaluation

We evaluate IDEM on DéjàKG through held-out policy comparisons, development-set sensitivity analyses, and selected extraction and integration cases.

6.1 Questions and experimental setup

The evaluation addresses three questions: (i) how do consolidation and extension outcomes vary across controlled categories; (ii) how do IDEM's representations and decision policies affect count-based update success; and (iii) how do count-based outcomes compare with trace-based semantic analysis?

The held-out evaluation compares IDEM's vector, LLM-based, and hybrid policies. Cross-framework evidence is limited to supplementary development comparisons with iText2KG (Appendix C.2); these do not establish held-out superiority over other frameworks. Table 1 gives the fixed test settings.

The IDEM sweeps use 5,000 Construction inputs and 500 probes per task. We hold two-stage extraction outputs and embeddings fixed to avoid variation from repeated extraction. Each configuration builds its own graph in the same input order. IDEM concatenates two weighted 3,072-dimensional views into a 6,144-dimensional vector. Appendix C.1 gives the tuning details.

The vector policy has four tuning parameters: two representation weights and two similarity thresholds. To limit the cost of searching them jointly, we used sequential development-set tuning. A small preliminary exploration guided the choice of entity/fact thresholds .75/.83 as starting values. We first

Table 1: Fixed IDEM test-set settings. Each description weight equals one minus its corresponding name or predicate-surface weight.

(a) Model settings

Setting	Value
LLM	`gpt-5.6-luna`
Reasoning effort	`low`
Embedding	`text-embedding-3-large`
Dimensions	3,072 per view

(b) IDEM parameters

Parameter	Value
Entity-name weight	.40
Predicate-surface weight	.00
Vector entity / fact thresholds	.70 / .70
Retrieval k (LLM-based / hybrid)	3
Hybrid fact threshold	.92
Hybrid fact margin	.05
Hybrid automatic entity matching	Disabled

swept the weights at these thresholds, then swept the thresholds at the selected weights. The hybrid policy shares the selected weights, with automatic entity matching disabled and a fact-reuse margin of .05. We evaluated nine candidate fact-reuse thresholds on the development set and selected .92, the smallest threshold tied for the highest VU.

All three policies use the same frozen test dataset, with settings fixed beforehand.

6.2 HELD-OUT TEST PERFORMANCE

Table 2 reports the completed IDEM test runs using the development-selected settings. All three policies share the same extraction outputs and are evaluated on 1,000 Construction inputs and 250 probes per task. We report Construction successes/total cases and probe successes/eligible cases; Appendix C.5 provides the category breakdown.

Table 2: Held-out IDEM test-set results (%). Parentheses show successes/total cases for Construction and successes/eligible probes for ECR/EER. Each probe task contains 250 cases before filtering. VU is the harmonic mean of the three unrounded scores.

Method	C	ECR	EER	VU
IDEM vector	97.00 (970/1000)	48.37 (119/246)	55.74 (136/244)	61.32
IDEM LLM-based	97.50 (975/1000)	91.46 (225/246)	87.24 (212/243)	91.88
IDEM hybrid	97.60 (976/1000)	93.50 (230/246)	85.25 (208/244)	91.82

6.3 REPRESENTATION AND POLICY ANALYSIS

The 121-setting weight sweep selected entity-name/predicate-surface weights .40/.00, with development VU of 68.19% at the initial .75/.83 thresholds. The subsequent 121-setting threshold sweep selected .70/.70, raising VU to 74.58%. ECR increased from 44.72% to 64.20%, while EER decreased from 87.25% to 69.88% and C from 98.34% to 96.74%. The selected setting therefore balances the three scores rather than maximizing each separately. Appendices C.1–C.2 give the full grids and component counts.

With these development-selected weights held fixed, test VU is 61.32% for vector matching, 91.88% for LLM-based matching, and 91.82% for hybrid matching. Hybrid has higher ECR but lower EER than the LLM-based policy. Their similar aggregate scores thus conceal different consolidation and extension outcomes; they do not establish equivalent decisions or policy-specific optimality.

Increasing entity-name weight helps with new facts about known entities, but hurts recognition of repeated facts. At the initial .75/.83 thresholds, raising the weight from .40 to 1.00 gives 41 more count-based successes on the former and 23 fewer on the latter, using probes eligible under both settings. Appendix C.3 details this trade-off.

Descriptions provide additional identity cues, but do not ensure that similarity separates equivalent from distinct facts. In the development examples (Appendix C.3), equivalent `works_as`/`has_profession` relations have surface/description cosines of .4568/.8622,

whereas opposing `is_above`/`is_below` relations score .7856/.9102. Because the opposing pair scores higher in both views, no shared nonnegative weighting and common similarity threshold can accept only the equivalent pair. For these generated representations, adjusting weights and thresholds is insufficient. These cases motivate supplementing vector similarity with LLM-based or hybrid decisions, rather than establishing an inherent limitation of all vector representations.

On the same eligible test-set probes, vector mainly shows excess growth in Consolidation and insufficient growth in Extension. LLM-based and hybrid matching reduce these predominant errors, though not every error type (Table 13, Appendix C.5).

6.4 Qualitative analysis of extraction and integration

Selected development traces distinguish extraction effects from integration decisions. A citizenship reformulation is extracted in a consistent direction and passes vector count checks, whereas a radius–diameter reformulation retains reversed endpoints and requires hybrid's LLM fallback. Hybrid also automatically merges an above/below contrast that the LLM-based policy keeps distinct. Appendix C.4 presents these IDEM cases; Appendix C.2 includes a baseline update with correct counts but inconsistent relation roles. Neither analysis estimates dataset-wide semantic error rates.

Extraction itself is another source of variation. We repeated extraction on 40 test inputs selected after inspecting the original outputs. The analysis distinguishes empty outputs and semantic errors from additional type facts that can be meaningful but violate the single-fact evaluation unit. Some original failures disappear on repetition; others persist. Appendix C.6 reports the selected-case study and its model-based assessment, separately from graph-update scores.

6.5 Scope and limitations

Count-based scoring supports reproducible evaluation without prescribing one lexical representation, but incorrect entities or links can satisfy expected counts. Conversely, meaning-preserving event nodes or multiple edges per claim can violate those counts. Shared output granularity is therefore required; qualitative inspection cannot eliminate these semantic blind spots.

Single-fact probes do not establish performance on passages with interacting claims, cross-sentence coreference, or context-dependent interpretations. This is an evaluation boundary, not evidence that a framework cannot process such inputs.

Without an additional within-input identity check, distinct mentions of one entity can receive separate new nodes, splitting an intended self-loop into an edge between duplicates. Identical mention keys or matching both mentions to the same existing node avoid this failure. Possible remedies include sequential resolution with intermediate updates or exact normalized-name deduplication within an input. Name-only merging risks conflating homonyms; neither remedy is evaluated here.

7 Conclusion

We present DéjàKG, a benchmark and count-based evaluation protocol for incremental knowledge graph construction, with VU summarizing Construction, Consolidation, and Extension. We also introduce IDEM, which combines identity-oriented descriptions, multi-view retrieval, and vector-based, LLM-based, or hybrid integration. Development analyses suggest complementary roles for names and descriptions: descriptions can distinguish same-name entities, while names can remain useful when descriptions vary across factual contexts. On the held-out test set, LLM-based and hybrid matching achieve similar VU scores, with higher consolidation but lower extension success for hybrid. These results describe count-based updates, not verified semantic accuracy. Improving description generation or developing alternative matching methods to reduce reliance on LLM decisions while preserving accuracy is a promising research direction for lowering KG construction cost and latency.

DéjàKG therefore complements benchmarks of semantic extraction quality. Combining its update-based protocol with checks of entity identity and relation meaning, while accommodating different extraction conventions, is another important research direction.

AI USE STATEMENT

The author defined the research problem, methodology, experimental design, dataset categories, and generation and review guidelines. Generative AI tools, including Codex and Grok, assisted with candidate-based sentence drafting and revision, implementation of the method and a review dashboard, identifying potential dataset errors, critical feedback on the design, literature search, and manuscript drafting and editing. The author iteratively reviewed dataset drafts and supplied correction feedback over several months. AI also assisted with code tests, numerical consistency checks, manuscript reviews, and analysis of development and test outputs. The extraction assessment in Appendix C.6 is model-based, not independent human annotation. The author retains responsibility for the submitted work.

REPRODUCIBILITY STATEMENT

The evaluation protocol and method are described in Sections 4 and 5, respectively. The dataset, IDEM implementation, count-based scoring code, and usage instructions are available in our anonymous repository: `https://anonymous.4open.science/r/dejaKG-2783`. The experimental settings are reported in the paper. Fresh runs require access to the specified model services and may yield different outputs; the released code does not guarantee exact reproduction of the reported numerical results.

REFERENCES

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. Leveraging linguistic structure for open domain information extraction. In *Proceedings of ACL-IJCNLP (Volume 1: Long Papers)*, pp. 344–354, 2015. URL https://aclanthology.org/P15-1034/.

Sangnie Bhardwaj, Samarth Aggarwal, and Mausam. CaRB: A crowdsourced benchmark for open IE. In *Proceedings of EMNLP-IJCNLP*, pp. 6262–6267, 2019. URL https://aclanthology.org/D19-1651/.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The Faiss library. *IEEE Transactions on Big Data*, 12(2):346–361, April 2026. ISSN 2372-2096. doi: 10.1109/TBDATA.2025.3618474. URL https://doi.org/10.1109/TBDATA.2025.3618474.

Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann. Knowledge graphs. *ACM Computing Surveys*, 54(4):71:1–71:37, 2021. doi: 10.1145/3447772. URL https://doi.org/10.1145/3447772.

Pere-Lluís Huguet Cabot and Roberto Navigli. REBEL: Relation extraction by end-to-end language generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2021. URL https://aclanthology.org/2021.findings-emnlp.204/.

Chengyue Jiang, Yong Jiang, Weiqi Wu, Yuting Zheng, Pengjun Xie, and Kewei Tu. COMBO: A complete benchmark for open KG canonicalization. In Andreas Vlachos and Isabelle Augenstein (eds.), *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 340–357, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.eacl-main.26. URL https://aclanthology.org/2023.eacl-main.26/.

Nora Kassner, Fabio Petroni, Mikhail Plekhanov, Sebastian Riedel, and Nicola Cancedda. EDIN: An end-to-end benchmark and pipeline for unknown entity discovery and indexing. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 8659–8673, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.593. URL https://aclanthology.org/2022.emnlp-main.593/.

Yassir Lairgi, Ludovic Moncla, Rémy Cazabet, Khalid Benabdeslem, and Pierre Cléau. iText2KG: Incremental knowledge graphs construction using large language models. In Mahmoud Barhamgi, Hua Wang, and Xin Wang (eds.), *Web Information Systems Engineering – WISE 2024*, pp. 214–229, Singapore, 2025. Springer Nature Singapore. ISBN 978-981-96-0573-6. doi: 10.1007/978-981-96-0573-6_16. URL https://link.springer.com/chapter/10.1007/978-981-96-0573-6_16.

Yassir Lairgi, Ludovic Moncla, Khalid Benabdeslem, Rémy Cazabet, and Pierre Cléau. ATOM: AdapTive and OptiMized dynamic temporal knowledge graph construction using LLMs. In Vera Demberg, Kentaro Inui, and Lluís Marquez (eds.), *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 950–966, Rabat, Morocco, March 2026. Association for Computational Linguistics. ISBN 979-8-89176-386-9. doi: 10.18653/v1/2026.findings-eacl.49. URL https://aclanthology.org/2026.findings-eacl.49/.

Nandana Mihindukulasooriya, Sanju Tiwari, Carlos F. Enguix, and Kusum Lata. Text2KGBench: A benchmark for ontology-driven knowledge graph generation from text. In Terry R. Payne, Valentina Presutti, Guilin Qi, María Poveda-Villalón, Giorgos Stoilos, Laura Hollink, Zoi Kaoudi, Gong Cheng, and Juanzi Li (eds.), *The Semantic Web – ISWC 2023*, pp. 247–265, Cham, 2023. Springer Nature Switzerland. ISBN 978-3-031-47243-5. doi: 10.1007/978-3-031-47243-5_14. URL https://link.springer.com/chapter/10.1007/978-3-031-47243-5_14.

George A. Miller. WordNet: a lexical database for english. *Communications of the ACM*, 38 (11):39–41, November 1995. ISSN 1557-7317. doi: 10.1145/219717.219748. URL https://doi.org/10.1145/219717.219748.

Belinda Mo, Kyssen Yu, Joshua Kazdan, Proud Mpala, Lisa Yu, Charilaos Kanatsoulis, and Sanmi Koyejo. KGGen: Extracting knowledge graphs from plain text with language models. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 30092–30115. Curran Associates, Inc., 2025. doi: 10.52202/085713-1010. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/2b368455e832d2b1a60bcad8c4c6481f-Paper-Conference.pdf.

Kim Anh Nguyen, Sabine Schulte im Walde, and Ngoc Thang Vu. Integrating distributional lexical contrast into word embeddings for antonym-synonym distinction. In *Proceedings of ACL (Volume 2: Short Papers)*, pp. 454–459, 2016. URL https://aclanthology.org/P16-2074/.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of ACL*, pp. 4902–4912, 2020. URL https://aclanthology.org/2020.acl-main.442/.

Shikhar Vashishth, Prince Jain, and Partha Talukdar. CESI: Canonicalizing open knowledge bases using embeddings and side information. In *Proceedings of the 2018 World Wide Web Conference*, 2018. doi: 10.1145/3178876.3186030. URL https://malllabiisc.github.io/publications/papers/cesi_www18.pdf.

Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, September 2014. ISSN 1557-7317. doi: 10.1145/2629489. URL https://doi.org/10.1145/2629489.

Bowen Zhang and Harold Soh. Extract, define, canonicalize: An LLM-based framework for knowledge graph construction. In *Proceedings of EMNLP*, pp. 9820–9836, 2024. URL https://aclanthology.org/2024.emnlp-main.548/.

A Dataset Design and Authoring

A.1 Shared procedure and evidence scope

Each category specifies an identity to preserve or a distinction to retain. The author screened candidate names, lexical relations, and source facts against category-specific criteria before AI-assisted sentence generation and iterative review. Existing Construction inputs are also transformed, and source-derived facts may undergo relation-specific sentence rewriting. The resources listed below describe the archived candidate bank, not the provenance of every final example.

All categories require one intended fact per sentence, sufficient context to resolve the tested ambiguity, consistent argument roles, and the intended identity relationships with the whole Construction graph. Added context must not introduce extra facts. These are authoring criteria, not a certification that every released example satisfies them. We retain reference entity and fact annotations to support future semantic evaluation beyond the count-based protocol.

Authored fictional entities. Fictional entities are intended to test integration of information supplied in the input, beyond established real-world facts. For example:

> *Construction:* The Northwind anthology includes the chapter Night Signals.
> *Extension (antonym):* The Northwind anthology omits the chapter Night Signals.

The intended update reuses both entities and adds a distinct claim. This is an input example, not a system result; fictional status does not establish absence from model training data.

LLM-assisted review. Review instances receive bounded sets of inputs, annotations, category criteria, and linked Construction anchors. They flag potential single-fact, alignment, identity, category, and direction errors; cross-case checks require retrieving additional examples. A coordinating agent re-examines flagged cases, and the author decides whether to accept, revise, or reject them. Model agreement is not human inter-annotator agreement or proof of correctness. This curation process is separate from benchmark scoring and does not establish complete release-review coverage.

A.2 Consolidation categories

Acronym, alias, and multilingual probes either restate an existing claim or introduce a new claim and opposite endpoint while reusing the designated entity. Inverse and relation-synonym probes preserve the complete claim.

Acronym. An abbreviation can have little lexical overlap with its full name, while the same abbreviation may denote different entities. These probes test whether the system recognizes the intended expansion rather than merging names unconditionally. Candidate leads include Wikidata short names (`P1813`). The author substitutes an established abbreviation and checks that the context supports the same referent; invented or unresolved abbreviations are excluded.

Alias. These probes test entity reuse under same-language names with little spelling overlap. Candidates include English Wikidata alternative labels (`skos:altLabel`) and personal pseudonyms (`P742`); real, pen, and stage names form part of alias, not a separate category. Authoring changes the name while preserving the referent. Review checks contextual identity and, for historical names, temporal compatibility. Association with a famous person does not establish identity; names of related entities, predecessors, or subunits are not automatically aliases.

Multilingual. These probes test reuse across language-specific names and scripts, including Quran/القرآن. Candidate leads include Wikidata native labels (`P1705`); authoring also replaces names in existing Construction inputs. It need not translate the whole sentence. Review requires established names of the same referent, excluding distinct language editions or similarly named subunits.

Inverse. Reversing arguments together with an inverse predicate can preserve a claim, whereas reversing arguments alone need not. These probes test whether integration respects both relation

meaning and argument roles. Authoring rewrites a Construction sentence in the inverse direction; Wikidata inverse-property links (P1696) provide additional leads. Review checks grammatical wording and equivalence of the complete assertion. An ordered-pair restriction can miss valid inverses, while ignoring roles can admit incorrect matches.

Relation synonym. Different predicate wordings can express the same claim, but a shared topic or one-way entailment does not establish equivalence. Candidate vocabulary includes sense-specific WordNet verb synsets. Authoring replaces the predicate while retaining endpoint order and roles. Review checks equivalence in context, including argument structure, modality, and aspect; synset membership alone is insufficient.

A.3 Extension categories

Homonym. Identical names can denote different entities, making surface equality insufficient for reuse. Dictionary-derived seeds, WordNet noun senses, and Wikipedia disambiguation links provide candidates. Authoring places the different referent in a single-fact context that identifies its intended sense. Review excludes unresolved ambiguity and dictionary-style statements that directly supply the distinction. The category tests referential ambiguity, not an etymological classification of homonymy versus polysemy.

Near entity. Similar but non-identical names can refer to distinct entities rather than aliases. Candidate discovery pairs Wikipedia titles linked to different Wikidata identifiers, using substring and spelling similarity as filters. Authoring states a fact about the distinct entity. Review checks contextual identity and absence of the new referent from Construction; different identifiers alone do not establish that distinction.

Sibling. These probes add a compatible fact with the same subject and predicate but a different object, testing whether integration preserves multiple values. Candidates come from source facts sharing a subject and predicate, supplemented by author-curated examples. One fact enters Construction and the other becomes the probe. Review checks that both can hold together and that the objects are distinct, rather than aliases or time-dependent replacements. Object-node creation depends on its presence in the full Construction graph.

Antonym. Opposite predicates can share distributional contexts and have similar embeddings (Nguyen et al., 2016). These probes test whether integration preserves a distinct claim despite that similarity. Dictionary-based opposition seeds and direct WordNet antonym links guide predicate substitution with ordered endpoints fixed. Review checks opposition in context and excludes equivalent argument-swapped inverses or mere insertion of *not*. Preserving a distinct assertion does not endorse its truth.

Different value. A substituted value changes a claim even when the subject and predicate remain identical. The author selects Construction sentences and replaces a value; the candidate bank also contains alternative values observed under the same predicate. Review checks an appropriate value type and a claim absent from Construction. Unlike sibling, the two values need not describe simultaneously true facts: the task tests preservation of distinct assertions, not factual correction.

A.4 Category overview and representative examples

Table 3 pairs each category's diagnostic purpose with a complete Construction/probe example from the working data inspected on September 19, 2026. These illustrate intended updates, not audited release examples or system outputs. Acronym and multilingual illustrate entity reuse within a new claim; antonym and different-value probes assert distinct claims, not factual corrections.

Table 3: Category purposes and representative input pairs. Construction supplies the prior information; each probe is evaluated independently against the resulting graph.

Category	What the probe tests	Construction input	Probe input
Consolidation: reuse equivalent information			
Acronym	Reuse an entity under an established abbreviation	The Apollo program was operated by the National Aeronautics and Space Administration.	The Space Shuttle program was managed by NASA.
Alias	Reuse an entity despite different names	Rudyard Kipling was born in Mumbai.	Rudyard Kipling was born in Bombay.
Multilingual	Reuse an entity across languages or scripts	Al-Fatiha is the first chapter of the Quran.	Arabic is the language of القرآن.
Inverse	Reuse a claim with reversed arguments and an inverse predicate	The radius is half the diameter.	The diameter is twice the radius.
Relation synonym	Reuse a claim under equivalent predicate wording	A thermostat regulates room temperature.	A thermostat controls room temperature.
Extension: preserve confusable new information			
Homonym	Separate different entities with identical names	Ankara is in Turkey.	Turkey is eaten on Thanksgiving.
Near entity	Separate different entities with similar names	Jordan is a country.	Michael Jordan played for the Chicago Bulls.
Sibling	Retain a compatible fact with a different object	Titanic features Leonardo DiCaprio in its cast.	Titanic features Kate Winslet in its cast.
Antonym	Separate opposing predicates with fixed argument order	The attic is above the cellar.	The attic is below the cellar.
Different value	Retain a claim with a substituted value	Tokyo Skytree is 634 metres tall.	Tokyo Skytree is 635 metres tall.

B Evaluation and Implementation Details

B.1 Operational count checks

We count facts extracted from input i before fact deduplication. Set $Q_i = 1$ if this count is exactly one and a fact-level output is recorded; otherwise, set $Q_i = 0$. This check does not require a new graph fact: an extracted fact may be identified as a duplicate and reused. Nor does it establish semantic correctness. For Construction input j, v_j^* and f_j^* denote the expected numbers of entities and facts newly added to the graph. We check these two counts separately:

$$b_j^V = Q_j \mathbf{1}[\Delta V_j = v_j^*], \qquad b_j^F = Q_j \mathbf{1}[\Delta F_j = f_j^*]. \tag{7}$$

The current entity-disjoint design sets $(v_j^*, f_j^*) = (2, 1)$. The scorer also supports older snapshots with legitimate endpoint reuse; their expectations must not be silently replaced by this design. For

probe i, let R_i^V and R_i^F contain the originating Construction cases for its required entities and facts. Then

$$B_i = \prod_{j \in R_i^V} b_j^V \prod_{j \in R_i^F} b_j^F. \tag{8}$$

The required knowledge comprises the designated anchor components and any entities or facts to be reused, not necessarily the entire anchor claim. The scorer derives these requirements and expected counts from the dataset's entity/fact IDs and anchor links; it does not align those IDs with system-output identities. An entity check is performed at its originating case's entity-count level; it is not a separate semantic match for that endpoint. This distinction matters when a case introduces two entities but only one is needed by a probe.

The recorded output depends on the adapter: IDEM traces expose fact decisions, and the legacy iText2KG adapter exposes extracted relationships. The scorer uses these respective records to measure the extracted fact count. Neither requirement certifies content, and successful reuse need not add a new edge. Complete observations and matching input hashes are required before scoring; an incomplete run is not a completed low-scoring run.

B.2 Extraction output constraints and failure handling

The first-stage prompt preamble and schema field descriptions are adapted from iText2KG. The output schema retains only subjects, predicates, and objects, omitting entity-category labels and temporal validity fields. Instructions referring to removed fields or an external entity extractor are also omitted. Additional guidance covers direction, output granularity, and preservation of stated relations and date values. Attribution and licensing notices are retained in the implementation.

The description stage receives the source text and fixed triples with identifiers for each subject/object slot. One structured response assigns each slot to a document-local referent group with a shared description and supplies a directed relation description for every triple. Entity instructions ask the model to use source context to distinguish intended referents, group mentions of the same entity, and keep homonyms separate. They favor stable identifying information; when an adequate definition is unavailable, the description may also state the current relation and counterpart. Relation descriptions must be source-grounded and explicitly preserve `SUBJECT`/`OBJECT` roles. Entity descriptions do not have the same explicit source-only restriction and are not verified attributes.

Validation checks complete, nonduplicated slot and triple-index coverage, nonempty descriptions, and both relation-role tokens. Descriptions are attached to the original triples, preserving their content, order, and direction. These checks enforce output structure, not semantic correctness. A contract violation triggers another request with the validation error, for at most three attempts. If all fail, the system supplies occurrence-specific unresolved entity descriptions and places the original predicate between `SUBJECT` and `OBJECT` as a relation template. This fallback does not guarantee semantic fidelity.

C Additional Experimental Analyses

C.1 Development weight sensitivity

The development sweep used 5,000 Construction, 500 Consolidation, and 500 Extension inputs. We held the two-stage extraction outputs (triples and descriptions) and their `text-embedding-3-large` embeddings fixed across configurations to avoid confounding parameter effects with variation from repeated LLM extraction. The extractor used `gpt-5.6-luna` with low reasoning effort. Each configuration built its own graph from an empty state using the same input order. We first fixed entity and fact similarity thresholds at .75 and .83 and varied entity-name and predicate-surface weights independently from 0 to 1 in increments of .1, giving 121 combinations. Description weights are their respective complements. We selected the combination with the highest development VU, then varied thresholds while holding these weights fixed (Appendix C.2). This is sequential parameter selection, not joint optimization of weights and thresholds.

We selected .40/.00 by unrounded VU, using Construction score and then lower lexical weights to break ties. Table 5 reports component maxima and the description-only/lexical-only endpoints.

Table 4: Development VU (%) across all 121 weight settings. Rows give entity-name weights; columns give predicate-surface weights. Each description weight is one minus the corresponding surface weight. Entity/fact thresholds are fixed at .75/.83. VU is calculated from unrounded component scores; bold marks the best setting.

	Predicate-surface weight										
Name wt.	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
0.0	61.64	61.44	60.88	60.41	59.66	58.84	57.99	56.82	54.47	53.90	53.31
0.1	64.60	64.55	63.88	63.41	62.66	61.84	60.84	59.66	57.33	56.76	56.18
0.2	67.00	66.95	66.28	65.66	64.91	64.10	63.10	61.94	59.44	58.87	58.49
0.3	68.14	67.96	67.42	66.94	66.17	65.33	64.31	63.11	60.54	59.96	59.77
0.4	**68.19**	68.00	67.43	66.94	66.14	65.29	64.27	63.04	60.40	59.80	59.60
0.5	66.56	66.37	65.79	65.29	64.48	63.61	62.58	61.32	58.83	58.22	58.02
0.6	65.81	65.63	65.06	64.57	63.73	62.87	61.85	60.62	58.15	57.55	57.35
0.7	65.67	65.52	64.71	64.24	63.45	62.64	61.69	60.53	58.40	57.83	57.64
0.8	65.75	65.61	64.83	64.38	63.61	62.82	61.90	60.77	58.88	58.34	58.15
0.9	66.77	66.67	65.95	65.39	64.67	63.93	63.07	62.02	60.26	59.74	59.57
1.0	66.57	66.48	65.65	65.24	64.54	63.96	63.13	62.12	60.58	60.09	59.92

Table 5: Best-scoring settings for each metric in the development weight grid, alongside description-only and surface-only settings. Scores are percentages; parentheses show successes/total cases for C and successes/eligible probes for ECR/EER. Each probe task contains 500 cases. Bold marks grid-wide maxima; tied settings are discussed in the text. VU uses unrounded scores.

Name wt.	Predicate-surface wt.	C	ECR	EER	VU
0.4	0.0	98.34 (4917/5000)	44.72 (220/492)	87.25 (431/494)	**68.19**
0.3	0.0	**98.46** (4923/5000)	45.23 (223/493)	85.02 (420/494)	68.14
1.0	0.0	93.32 (4666/5000)	**48.35** (235/486)	73.15 (346/473)	66.57
0.4	0.8	98.34 (4917/5000)	35.37 (174/492)	**89.07** (440/494)	60.40
0.0	0.0	98.02 (4901/5000)	41.38 (204/493)	69.92 (344/492)	61.64
1.0	1.0	93.32 (4666/5000)	38.48 (187/486)	74.84 (354/473)	59.92

Where a component maximum is tied, it shows the highest-VU representative: C is tied across predicate weights at entity weight .30, and EER across predicate weights .80–1.00 at entity weight .40.

Follow-up LLM-based and hybrid runs share weights .40/.00 and extraction outputs, but construct separate graphs. The initial hybrid run uses a fact threshold of .91 and margin .05, with automatic entity matching disabled. A separate threshold search subsequently selects .92; Table 7 reports that setting.

Zero predicate-surface weight removes that embedding channel, not the recorded predicate. Construction outcomes and probe eligibility can still vary with integration settings. Reusing embeddings controls one source of variation; it does not establish that the selected weights are optimal for the LLM-based or hybrid policies.

C.2 Threshold selection after fixing the weights

After selecting weights .40/.00, we varied the vector entity and fact thresholds independently over $\{.50, .55, \ldots, 1.00\}$, yielding the 121 threshold pairs reported in Figure 3. Both stages used the same fixed inputs, extraction outputs, and embeddings. The initial .75/.83 thresholds were fixed settings for weight selection, not independently established optima. We selected the threshold pair with the highest development VU in this grid.

The .70/.70 pair maximizes development VU at 74.58%, only 0.12 percentage points above .70/.75. Selection used unrounded VU, then Construction score and higher thresholds to break ties. This selects a measured grid setting, not a statistically established or continuous optimum.

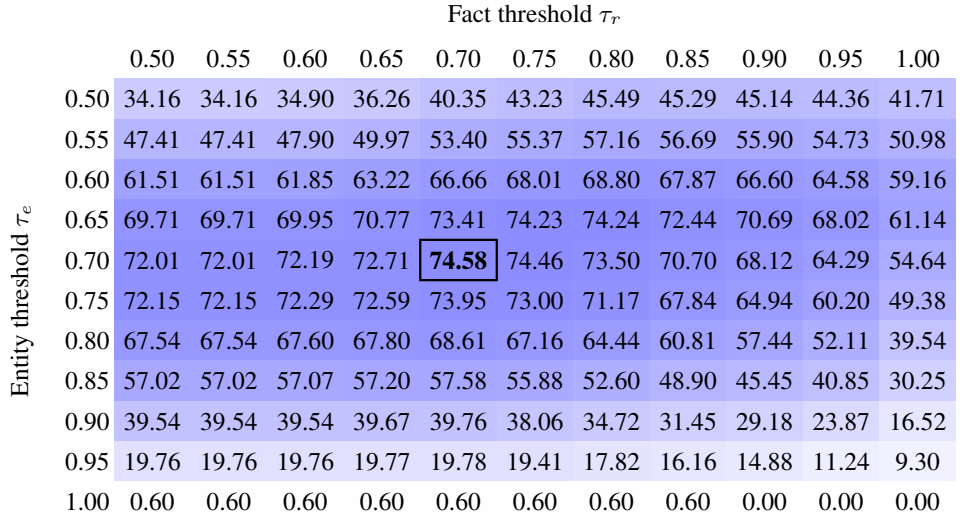

Figure 3: Development VU (%) across 121 entity/fact threshold pairs, with entity-name weight .40 and predicate-surface weight .00. Darker cells indicate higher VU; the outlined maximum uses both thresholds .70. Each setting builds its own Construction graph from the same fixed extraction outputs.

Table 6: Selected vector entity/fact thresholds (τ_e, τ_r) with entity-name/predicate-surface weights fixed at .40/.00. The initial .75/.83 setting is included for comparison with the grid's VU winner, runner-up, and component maxima. Scores are percentages; parentheses show successes/total cases for C and successes/eligible probes for ECR/EER. Each probe task contains 500 cases. Bold marks grid-wide maxima.

τ_e	τ_r	C	ECR	EER	VU
0.75	0.83	98.34 (4917/5000)	44.72 (220/492)	87.25 (431/494)	68.19
0.70	0.70	96.74 (4837/5000)	64.20 (312/486)	69.88 (341/488)	**74.58**
0.70	0.75	96.74 (4837/5000)	60.70 (295/486)	74.18 (362/488)	74.46
0.95	0.70	**99.66** (4983/5000)	8.25 (41/497)	48.99 (243/496)	19.78
0.55	0.60	68.00 (3400/5000)	**79.04** (328/415)	28.35 (110/388)	47.90
0.75	0.95	98.34 (4917/5000)	35.16 (173/492)	**89.07** (440/494)	60.20

Table 6 shows the trade-off: maximizing Construction leaves ECR low, while maximizing ECR sacrifices Construction and Extension. Relative to .75/.83, the selected pair improves Consolidation but lowers the other two scores. Tied component maxima are represented by their highest-VU setting: C ties at entity thresholds .95 and 1.00 across fact thresholds; ECR at entity threshold .55 with fact thresholds .50, .55, and .60; and EER at entity threshold .75 with fact thresholds .95 and 1.00.

Both sweeps guide sequential parameter selection on development data; they do not isolate either threshold's causal effect or establish held-out performance.

Development results at the selected settings. Table 7 provides a supplementary development comparison with the legacy iText2KG class at revision `c213cc3`, not `iText2KG_Star` (Lairgi et al., 2025). Its entity/relation thresholds are .70/.70 and entity-name/label weights .60/.40. IDEM uses weights .40/.00; vector thresholds are .70/.70, LLM-based and hybrid policies use $k = 3$, and hybrid uses fact threshold .92 and margin .05. All runs used `gpt-5.6-luna`. This comparison uses development data, not a held-out test set.

ATOM uses similar cosine-threshold resolution but also addresses document decomposition, temporal modeling, and parallel graph merging (Lairgi et al., 2026). Those capabilities are outside this single-fact protocol; the comparison does not imply equivalence between ATOM and iText2KG.

Table 7: Supplementary development results at the selected IDEM settings and iText2KG defaults. Scores are percentages. Parentheses give successes/total for C and successes/eligible for ECR/EER; each probe task contains 500 cases. VU uses the three unrounded rates.

System	C	ECR	EER	VU
iText2KG (native)	36.68 (1834/5000)	46.38 (128/276)	40.57 (86/212)	40.83
IDEM vector	96.74 (4837/5000)	64.20 (312/486)	69.88 (341/488)	74.58
IDEM LLM-based	99.14 (4957/5000)	94.14 (466/495)	91.79 (447/487)	94.92
IDEM hybrid	99.14 (4957/5000)	94.55 (468/495)	91.79 (447/487)	95.06

Over-merging during Construction. iText2KG's low Construction score does not primarily reflect empty or multi-fact extractions: 4,402 of 5,000 inputs yield exactly one extracted fact. Of these, 2,568 add fewer than the expected two entities, while the remaining 1,834 pass the Construction check. Inspection confirms incorrect entity reuse in individual cases. For "Josef Schäffer is a citizen of Austria," iText2KG extracts Austria correctly but stores a citizenship edge to the existing Germany node. Similarly, "Pati Patni Aur Tawaif originated in India" yields India during extraction but Pakistan in the stored edge. Each update adds one entity and one fact instead of two entities and one fact. Construction therefore tests integration into the growing graph, not extraction alone. These cases establish that over-merging contributes to the low score; they do not classify all 2,568 low-growth cases as incorrect merges.

A separate pairwise recomputation with 3,072-dimensional `text-embedding-3-large` vectors helps explain these substitutions. Austria–Germany and India–Pakistan have name-only cosines of .5499 and .5350; both pairs share the label `Country`, giving a label-only cosine of 1.0000. Under iText2KG's vector combination, $0.6\, v_{\text{name}} + 0.4\, v_{\text{label}}$, the corresponding cosines rise to .7543 and .7469, exceeding the .70 entity threshold. These are freshly recomputed pair scores, not recovered historical candidate scores or a replay of the full candidate ranking.

Correct counts can hide inconsistent relation roles. The probe "The pound sterling is the currency of the UK" should reuse the country introduced by "Romany Love originated in the United Kingdom" and add one currency entity and one fact. iText2KG adds exactly these counts and passes the count check. However, it replaces the extracted predicate `is_currency_of` with `used_currency` while retaining the currency-to-country endpoint order. The stored probe edge contrasts with an existing edge extracted from "The currency of British North Borneo was the British North Borneo dollar":

Subject	Predicate	Object
British North Borneo	`used_currency`	British North Borneo dollar
pound sterling	`used_currency`	United Kingdom

The existing edge uses `used_currency` from region to currency, whereas the probe edge uses it in the opposite direction. The count check cannot detect this inconsistency. This case illustrates a semantic error hidden by correct counts, not its frequency across the dataset.

C.3 WEIGHT SENSITIVITY AND REPRESENTATION ANALYSIS

We examine how entity-name and predicate-surface weights affect each category. Starting from weights .40/.00, we vary one weight at a time while holding the other fixed (Figure 4). These analyses use thresholds .75/.83, as in the weight-selection sweep, rather than the subsequently selected .70/.70 thresholds. They therefore describe weight sensitivity under the original thresholds, not the final configuration.

Different weights can produce different Construction graphs, so the probes eligible for evaluation can also change. Figure 4 reports ECR/EER using all eligible probes at each setting. Figure 5 instead compares the same probes across settings: it retains only those eligible under all 11 tested values of the varied weight. This fixes which cases are compared, but each setting still uses its own Construction graph.

Table 8 reports the corresponding probe counts. For example, 85–99 different-value probes are eligible at individual entity-name weights, but only 83 are eligible at every weight. Figure 5 evaluates those same 83 probes throughout. These common-set success rates supplement, rather than replace, ECR/EER. In both figures, an eligible probe whose extraction fails is counted as a failure.

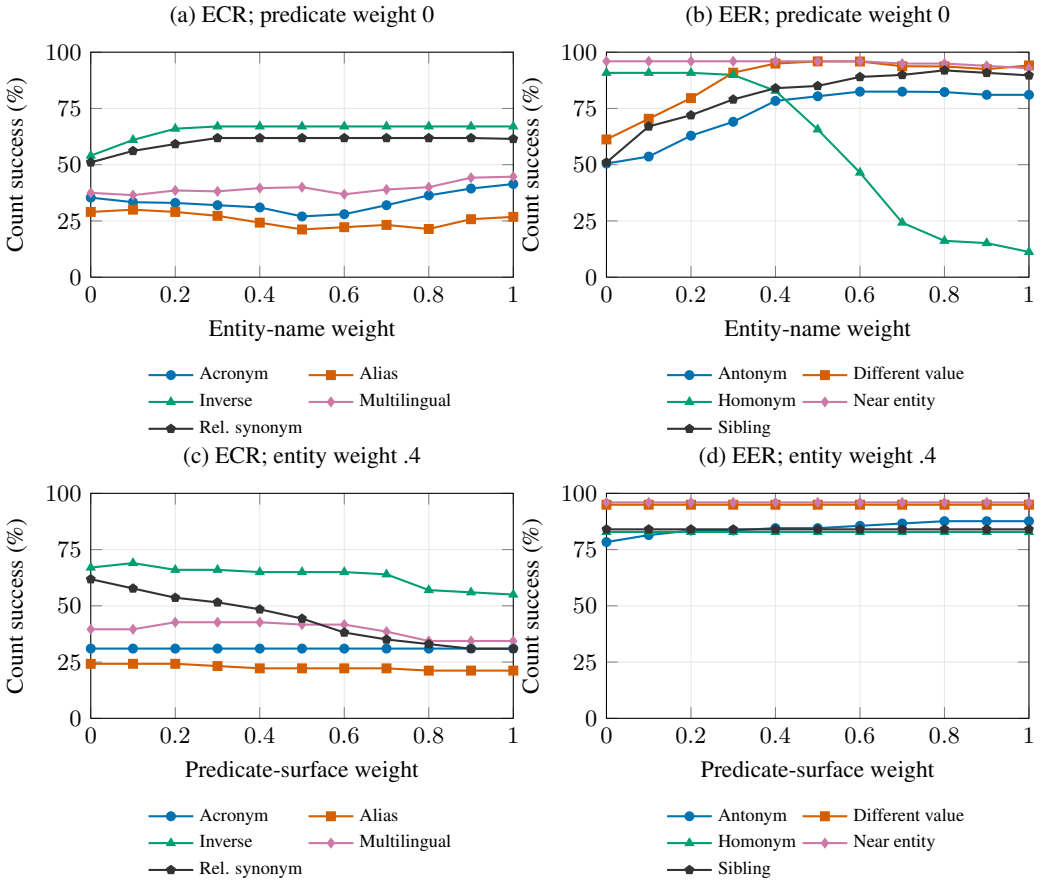

Figure 4: Category scores as one weight varies, using the probes eligible at each setting. The evaluated cases can therefore change across points. Solid curves show ECR/EER; each category contains 100 probes before filtering. Entity/fact thresholds are .75/.83. Lines connect measured settings, not additional measurements. Figure 5 instead compares the same cases across settings.

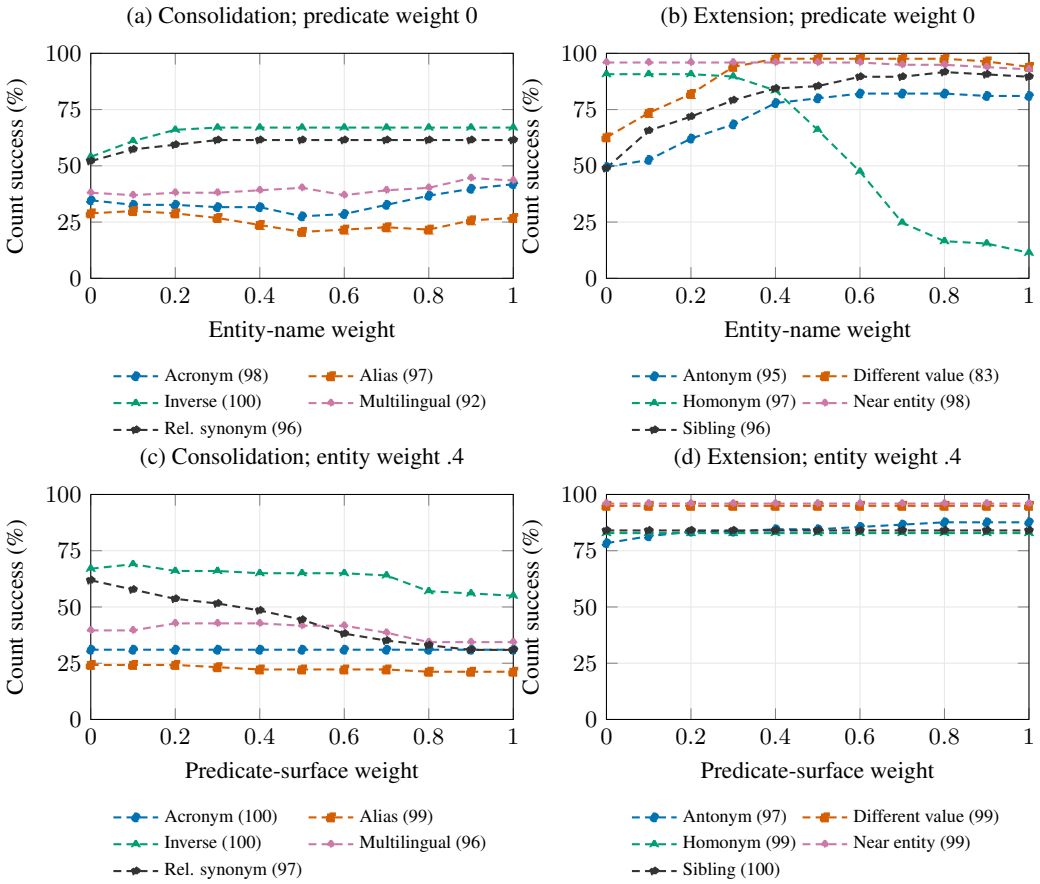

Figure 5: Category scores using the same probes across all 11 tested weights. For each one-weight sweep, only probes eligible at every setting are retained; legend parentheses give their counts. Dashed curves show supplementary count-success rates, not replacement ECR/EER scores. Panels, axes, colors, and thresholds match Figure 4. Lines connect measured settings.

Table 8: Probe counts for the weight curves. Common n counts cases eligible at all 11 weights, used throughout Figure 5. Eligible range gives the smallest and largest case counts across Figure 4 settings. Each category starts with 100 probes. Table 9 instead compares cases eligible under two settings.

		Entity-name weight sweep		Predicate-surface weight sweep	
Task	Category	Common n	Eligible range	Common n	Eligible range
Consolidation	Acronym	98	99–100	100	100–100
	Alias	97	97–100	99	99–99
	Inverse	100	100–100	100	100–100
	Multilingual	92	94–97	96	96–96
	Relation synonym	96	96–98	97	97–97
Extension	Antonym	95	95–97	97	97–97
	Different value	83	85–99	99	99–99
	Homonym	97	98–99	99	99–99
	Near entity	98	98–99	99	99–99
	Sibling	96	97–100	100	100–100

Table 10 accompanies the category-level analysis with six paired examples, showing inputs, generated descriptions, similarities, and update outcomes together. The sweep records establish count outcomes but do not identify the selected nodes and edges.

Descriptions help distinguish identical names. The weight curves show why neither view is sufficient on its own. On the fixed set of 97 homonym probes, increasing entity-name weight from zero to one reduces count successes from 88 to 11. The Turkey pair illustrates the missing information: the same name denotes a country in "Ankara is in Turkey" and food in "Turkey is eaten on Thanksgiving." The descriptions generated from these inputs make this distinction explicit: "Turkey, the country in which Ankara is located" versus "Turkey, the food eaten on Thanksgiving." Their name cosine is 1.0000, whereas their description cosine is .5111. Combining the views lowers this intended-pair score to .7067, below the .75 entity threshold. The combined setting passes the count check, while the name-only setting does not (case 3, Table 10). This links a concrete sense distinction to the observed scores, without verifying which node the system selected.

Names remain useful when descriptions vary across facts. Descriptions also vary across facts about the same entity. The Construction sentence "J. Edgar Hoover worked for the Federal Bureau of Investigation" yields "The Federal Bureau of Investigation, the organization for which J. Edgar Hoover worked." The probe "The Department of Justice supervises the FBI" instead yields "The FBI, the Federal Bureau of Investigation supervised by the Department of Justice in the source text." Both descriptions identify the same organization but include different relationships. The Austria pair shows the same pattern: the description is "Austria, the country of which Josef Schäffer is a citizen." for the citizenship statement, whereas "The Alps extend across Österreich" yields "Österreich, the German name for Austria, the country across which the Alps extend." Although the probe descriptions explicitly identify the corresponding names, their cosines are only .5745 and .4987, below the name cosines of .8262 and .8363. The name-only setting passes both count checks; the combined setting adds an extra entity (cases 1–2, Table 10). Thus, making identity explicit in a description does not necessarily produce a higher embedding similarity.

Reusing entities in new facts differs from recognizing repeated facts. Consolidation tests both restatements of existing facts and new facts about existing entities. A restatement requires reusing the entities and the fact; a new statement, as in the FBI and Austria examples, requires reusing the designated entity while adding a new fact. Table 9 reports these settings separately. Among probes eligible under both compared configurations, increasing entity-name weight from .40 to 1.00 raises count-based successes on new-fact probes from 18/146 to 59/146. In contrast, successes on restatements fall from 199/340 to 176/340. Thus, the higher overall success count masks a decline on restatements, while success on new-fact probes remains below half. These counts describe complete updates, not entity-resolution accuracy alone.

Table 9: Count-based Consolidation successes under two weight changes. Each block compares the same probes under both settings, retaining only cases eligible in both.

Entity-name weight: .40 → 1.00 (predicate-surface weight fixed at .00)			
Input type	Evaluated probes	Successes at .40	Successes at 1.00
Existing fact expressed again	340	199	176
New fact about an existing entity	146	18	59
Predicate-surface weight: .00 → .80 (entity-name weight fixed at .40)			
Input type	Evaluated probes	Successes at .00	Successes at .80
Existing fact expressed again	344	202	156
New fact about an existing entity	148	18	18

Table 10: Development input pairs, generated descriptions, similarities, and observed updates. Descriptions are quoted verbatim. Similarities compare the displayed pairs, not necessarily the candidates selected by the system. Growth $(\Delta V, \Delta F)$ gives the numbers of added entities and facts; bold marks agreement with expected counts, not verified identities. Each case is eligible under both compared settings.

Construction	Probe
Entity-name weight: .40 → 1.00. Predicate-surface weight fixed at .00; entity threshold .75.	
Case 1. Acronym (Consolidation). Expected growth $(1, 1)$.	
"J. Edgar Hoover worked for the Federal Bureau of Investigation."	"The Department of Justice supervises the FBI."
Name: `Federal Bureau of Investigation`. *Definition:* "The Federal Bureau of Investigation, the organization for which J. Edgar Hoover worked."	*Name:* `FBI`. *Definition:* "The FBI, the Federal Bureau of Investigation supervised by the Department of Justice in the source text."
Cosine similarity: name .8262; description .5745.	
Name weight .40: weighted score .6752; observed growth $(2, 1)$.	*Name weight 1.00:* weighted score .8262; observed growth $(\mathbf{1}, \mathbf{1})$.
Case 2. Multilingual (Consolidation). Expected growth $(1, 1)$.	
"Josef Schäffer is a citizen of Austria."	"The Alps extend across Österreich."
Name: `Austria`. *Definition:* "Austria, the country of which Josef Schäffer is a citizen."	*Name:* `Österreich`. *Definition:* "Österreich, the German name for Austria, the country across which the Alps extend."
Cosine similarity: name .8363; description .4987.	
Name weight .40: weighted score .6338; observed growth $(2, 1)$.	*Name weight 1.00:* weighted score .8363; observed growth $(\mathbf{1}, \mathbf{1})$.
Case 3. Homonym (Extension). Expected growth $(2, 1)$.	
"Ankara is in Turkey."	"Turkey is eaten on Thanksgiving."
Name: `Turkey`. *Definition:* "Turkey, the country in which Ankara is located"	*Name:* `Turkey`. *Definition:* "Turkey, the food eaten on Thanksgiving."
Cosine similarity: name 1.0000; description .5111.	
Name weight .40: weighted score .7067; observed growth $(\mathbf{2}, \mathbf{1})$.	*Name weight 1.00:* weighted score 1.0000; observed growth $(1, 1)$.
Predicate-surface weight: .00 → .80. Entity-name weight fixed at .40; fact threshold .83.	
Case 4. Antonym (Extension). Expected growth $(0, 1)$.	
"The attic is above the cellar."	"The attic is below the cellar."
Predicate: `is_above`. *Template:* "SUBJECT is above OBJECT"	*Predicate:* `is_below`. *Template:* "SUBJECT is below OBJECT"
Cosine similarity: predicate .7856; description .9102.	
Predicate weight .00: weighted score .9102; observed growth $(0, 0)$.	*Predicate weight .80:* weighted score .8105; observed growth $(\mathbf{0}, \mathbf{1})$.
Case 5. Relation synonym (Consolidation). Expected growth $(0, 0)$.	
"Sam Milai worked as a cartoonist."	"Sam Milai worked professionally as a cartoonist."
Predicate: `works_as`. *Template:* "SUBJECT worked as OBJECT"	*Predicate:* `has_profession`. *Template:* "SUBJECT worked professionally as OBJECT"
Cosine similarity: predicate .4568; description .8622.	
Predicate weight .00: weighted score .8622; observed growth $(\mathbf{0}, \mathbf{0})$.	*Predicate weight .80:* weighted score .5379; observed growth $(0, 1)$.
Case 6. Inverse (Consolidation). Expected growth $(0, 0)$.	
"José González Ganoza is a citizen of Peru."	"Peru is José González Ganoza's country of citizenship."
Predicate: `is_citizen_of`. *Template:* "SUBJECT is a citizen of OBJECT"	*Predicate:* `has_country_of_citizenship`. *Template:* "SUBJECT has OBJECT as their country of citizenship"
Cosine similarity: predicate .7739; description .8408.	
Predicate weight .00: weighted score .8408; observed growth $(\mathbf{0}, \mathbf{0})$.	*Predicate weight .80:* weighted score .7873; observed growth $(0, 1)$.

Predicate weighting affects synonyms and antonyms differently. On the fixed category sets in Figure 5, increasing predicate-surface weight from zero to one reduces count-based successes on relation-synonym probes from 60/97 to 30/97, but increases successes on antonym probes from 76/97 to 85/97. The paired examples show how these opposing trends can arise. For the occupation pair, `works_as` and `has_profession` have a surface cosine of .4568, but their descriptions score .8622. At the .83 threshold, the description-only setting matches the expected zero-fact increment, whereas adding predicate-surface weight produces an extra fact. The citizenship pair follows the same pattern. Both citizenship extractions already point from the person to Peru, so this example concerns predicate wording, not matching reversed endpoints.

Above/below shows the opposite outcome. Its descriptions score .9102 despite expressing distinct claims, and the description-only setting adds no new fact. Greater predicate-surface weight lowers the score to .8105 and the count check passes. Changing a shared threshold cannot solve both this contrast and the occupation pair: above/below scores higher in both individual channels, so it also scores higher under any shared nonnegative weighting. A rule that accepts the occupation pair by exceeding that threshold must accept above/below as well. This limitation applies to these two representation pairs and this scalar similarity rule, not to every embedding or identity policy.

These representation contrasts do not isolate the effects of context, generation, or embeddings. To examine the actual matches rather than intended-pair similarities alone, the next section turns to recorded integration decisions.

C.4 Case analysis of IDEM retrieval and integration

We examine how IDEM distinguishes retrieved entity candidates and decides whether to reuse an existing fact. Each development example contrasts the expected update with the outcomes of the relevant IDEM policies.

Settings and evidence. IDEM variants share extraction outputs and weights .40/.00. Vector thresholds are .70/.70, rather than the .75/.83 used in the preceding weight analysis; hybrid uses a .91 fact gate with margin .05 and no automatic entity matching. LLM-based and hybrid decisions were checked against Construction nodes and edges. Vector evidence establishes count increments, not the selected identities. These selected, model-assisted cases do not estimate error frequencies or isolate individual mechanisms.

Retrieved candidates can refer to different entities. The graph already contains the city in "Hamilton is in the Waikato region." The incoming statement, "Hamilton was written by Lin-Manuel Miranda," refers to a musical and should not reuse that city. IDEM retrieves the city and Alexander Hamilton as candidates, but its LLM-based and hybrid policies reject these identities and create separate nodes for the musical and its author. This case illustrates why retrieval and identity decisions are distinct: a candidate can share a name without referring to the same entity.

LLM-based comparison can recover low-similarity matches. After "A capstan winds a rope," the statement "A capstan coils a rope" expresses the same fact and should reuse the existing edge. IDEM vector adds a fact, whereas IDEM's LLM-based and hybrid policies resolve the intended endpoints and reuse the edge. The latter policies' candidate traces report a relation similarity of .6090, yet LLM comparison recognizes the equivalent meaning. This score comes from those traces, not a recovered vector-policy comparison; the vector result establishes the extra fact, not its precise cause.

Argument direction matters for fact matching. Equivalent statements can produce triples with either the same or reversed endpoint order. Table 11 pairs the extracted triples with the expected action and observed outcomes. Its third case shows that matching direction alone is not enough: opposite claims can have the same endpoint order.

Case 1 is compatible with vector's same-order restriction: both extractions point from the person to Qatar, with identical descriptions, `SUBJECT is a citizen of OBJECT` (similarity 1.0000 in the LLM-based/hybrid traces). Its passing counts do not demonstrate inverse reasoning by the vector matcher; the extraction has already aligned the directions.

Case 2 retains opposite directions: `SUBJECT is half of OBJECT` and `SUBJECT is twice OBJECT`. Although their recorded similarity is .8157, vector excludes reverse-direction

Table 11: Relation direction and fact reuse in selected development cases. Both entities should be reused in every case. For vector, only graph-count changes were inspected; for LLM-based and hybrid matching, the selected nodes and edges were also checked against the intended identities.

Construction input and extraction	Probe input and extraction
1. Consistent direction: inverse wording	
Saad Hussain is a citizen of Qatar. `(Saad Hussain, is_citizen_of, Qatar)`	Qatar is Saad Hussain's country of citizenship. `(Saad Hussain, is_citizen_of, Qatar)`
Expected: reuse the fact. Observed: vector adds no entity or fact; LLM-based reuses the fact; hybrid reuses it automatically.	
2. Reversed direction: inverse wording	
The radius is half the diameter. `(radius, is_half_of, diameter)`	The diameter is twice the radius. `(diameter, is_twice, radius)`
Expected: reuse the fact. Observed: vector adds one fact and no entities; LLM-based and hybrid reuse the reversed edge through LLM comparison.	
3. Same direction, distinct claim: antonym	
The attic is above the cellar. `(attic, is_above, cellar)`	The attic is below the cellar. `(attic, is_below, cellar)`
Expected: add the distinct below fact. Observed: vector adds no fact; hybrid incorrectly reuses the above fact automatically; LLM-based adds the below fact.	

fact candidates and records an extra fact. Both LLM-based routes resolve the correct endpoints and recognize that the reversed half-of edge expresses the same claim. Thus, the extraction need not align directions for these policies to reuse the fact.

Case 3 exposes the limit of automatic reuse. The LLM-based and hybrid policies resolve attic and cellar correctly, but the relation similarity of .9102 exceeds hybrid's .91 gate. With only one fact candidate, the margin does not test separation from a competitor, and hybrid incorrectly reuses the above fact. The LLM-based policy instead preserves the distinct below claim. This failure concerns the reported gate, not every possible hybrid threshold: correct endpoints and high similarity do not by themselves establish claim equivalence.

C.5 Test results by category and growth-error direction

Table 12 decomposes the IDEM test results in Table 2. Each category contains 50 probes before filtering. Entries are successful/eligible cases under the conditional protocol; failed probe extractions remain failures when their Construction prerequisites are satisfied. Category totals recover the overall ECR/EER counts. These are count-based update outcomes, not isolated entity- or relation-resolution accuracies.

Table 12: IDEM test-set results by category. Entries show successful/eligible probes; each category contains 50 probes before filtering.

Category	Vector	LLM-based	Hybrid
Consolidation (ECR)			
Acronym	23/49	47/49	47/49
Alias	13/48	39/48	42/48
Inverse	23/50	49/50	48/50
Multilingual	21/50	47/50	48/50
Relation synonym	39/49	43/49	45/49
Extension (EER)			
Homonym	20/48	36/48	36/48
Antonym	16/50	44/50	41/50
Near entity	30/49	48/49	48/49
Sibling	38/49	44/49	41/49
Different value	32/48	40/47	42/48

Direction of count errors. To compare failure patterns on the same inputs, Table 13 retains only probes eligible under all three policies: 246 Consolidation and 243 Extension cases. These restricted counts supplement, rather than replace, the original ECR/EER results. We first separate inputs that fail the extraction check (exactly one extracted fact and an observed fact-level output). For the remaining inputs, we subtract the expected entity and fact increments from the observed increments. Both differences zero means count agreement. Nonnegative differences with at least one positive indicate excess growth; nonpositive differences with at least one negative indicate insufficient growth. Opposite signs define mixed growth errors; none occur in this comparison.

Table 13: Test-set outcomes for the same probes across all three policies: 246 Consolidation and 243 Extension cases whose Construction prerequisites hold for every policy. Each probe is counted in exactly one column. Cases that fail the extraction check appear only under extraction failure, not under growth errors.

Policy	Count agreement	Excess growth	Insufficient growth	Extraction failure
Consolidation: 246 common-eligible probes				
Vector	119	121	4	2
LLM-based	225	19	0	2
Hybrid	230	14	0	2
Extension: 243 common-eligible probes				
Vector	135	1	102	5
LLM-based	212	15	11	5
Hybrid	207	20	11	5

Vector failures predominantly involve excess growth in Consolidation and insufficient growth in Extension, rather than a uniform tendency toward insufficient growth. LLM-based and hybrid matching reduce these respective counts from 121 to 19/14 and from 102 to 11/11. However, Extension excess-growth cases increase from one for vector to 15/20, so improvement is not uniform across error directions. These are structural discrepancies, not verified missed or incorrect merges: extraction content and incorrect identities can also affect growth. Common probe membership does not make the Construction graphs identical or isolate the causal effect of the matching policy.

C.6 Extraction stability on selected failure candidates

We asked whether problematic extractions recur on repetition and whether changing the model or reasoning effort helps. We selected 40 test inputs from the original Luna-low run: 14 with empty outputs, 18 with multiple triples, six with a single self-loop, and two with direction errors. Using the same two-stage extractor and instructions, we repeated extraction three times under each of three settings: Luna low, Luna medium, and Terra low. The original run was used only to select cases and is excluded from the repeated-run results.

An output passed the assessment if it contained exactly one triple, preserved the intended entities, relation, direction, and scope, and had no conflict between its predicate wording and description. Equivalent inverse formulations were accepted. An AI review agent assessed all 360 outputs against the input, reference annotations, and linked Construction text. Table 14 separates the pass rate in each run from repeatability: the final column counts inputs that passed in all three runs, without requiring identical wording.

Table 14: AI-assessed extraction results on 40 selected failure candidates. A pass requires one triple and the semantic checks described above. Run columns show passed inputs/40; "Passed all 3" counts inputs passing every repeat. Mean and sample SD summarize the three run-level pass rates, not full-test accuracy or confidence intervals.

Model / effort	Run 1	Run 2	Run 3	Pass rate $\pm$ SD (%)	Passed all 3
Luna / low	17/40	17/40	21/40	45.8 ± 5.8	10/40
Luna / medium	21/40	16/40	18/40	45.8 ± 6.3	11/40
Terra / low	31/40	28/40	31/40	75.0 ± 4.3	24/40

Terra passed more inputs in every run and passed all three runs on 24 inputs, compared with 10 for Luna low and 11 for Luna medium. Luna's mean pass rate was unchanged by the higher reasoning effort. These observations suggest that changing the extraction model may improve some failures, while additional reasoning effort alone need not do so. The examples below distinguish missing information, additional facts, and conflicting relation roles.

Empty extraction. "The log records events" produced an empty initial triple list in all nine extractions. The failure occurred before description generation, rather than through removal of a previously extracted fact.

Additional facts. "The OSIRIS-REx spacecraft sampled asteroid Bennu" can yield the sampling relation plus type facts for spacecraft and asteroid. Such additions follow the prompt's instruction to extract stated types, but exceed the protocol's single-fact unit. Failing this constraint does not, by itself, make the additional facts false.

Conflicting relation roles. All three Terra outputs for "València hosted the 2007 America's Cup" used `(2007 America's Cup, hosts, València)` with the description "OBJECT hosted SUBJECT." The predicate says that the event hosts the city, while the description says that the city hosted the event. A single triple can therefore still contain a semantic inconsistency.

Scope of the analysis. These inputs were selected after inspecting Luna-low outputs, not sampled to represent the full test set. The assessment was model-based, with model and repetition labels visible; it was neither blinded nor independent human annotation. The results do not establish full-test accuracy, a general model ranking, or a causal effect of reasoning effort. They concern extraction only: the diagnostic outputs neither replaced the official test outputs nor underwent graph-update scoring, so this study cannot identify semantic false passes or false failures in the final graph evaluation.