Policy-Driven Contradiction Resolution for Temporal Knowledge Graphs

Kelvin Jou
University of California, Santa Barbara, USA
kelvinjou@ucsb.edu

***Abstract*—Temporal knowledge graphs must reconcile newly ingested evidence with existing relationships while preserving the history and provenance of prior claims. This process is difficult in practice because domains follow distinct conventions for determining whether competing claims should replace one another, coexist, remain unresolved, or represent a temporal state transition. We present an experimental policy layer for KG-Gen, a library that constructs knowledge graphs from natural-language documents. The layer allows domain experts to express conflict-resolution and observation-lifecycle policies in natural language. We evaluate the approach in the vaccine-guidance domain using source priorities and terminology informed by CDC and FDA guidance [2]. Our benchmark contains 75 synthetic temporal observations, including 36 controlled claim pairs spanning multiple epistemic relationships and 11 observation-expiration events. Compared with the policy-free run, the policy-conditioned LLM achieved higher policy-grounded decision accuracy (39.3% versus 25.0%), conditional policy-action accuracy among cases routed to a resolver (64.7% versus 38.9%), overall resolution accuracy (30.6% versus 19.4%), and lifecycle-resolution accuracy (72.7% versus 45.5%). However, policy-routing coverage decreased from 64.3% to 60.7%, while issue-type accuracy remained unchanged at 72.2%. These results suggest that natural-language policies can improve domain-specific resolution and lifecycle handling after a claim is routed to the appropriate resolver, but do not necessarily improve the initial classification or routing of that claim.**

I. Introduction

Knowledge graphs should not be append-only, especially temporal ones. Newly received evidence may update, contradict, or coexist with evidence already in the knowledge graph. When knowledge graphs naively accumulate every claim, they may introduce inconsistencies. However, automatically replacing an old claim with a new one may also erase historically valid information. In addition, temporal changes must be distinguished from contradictions while preserving the provenance of prior claims.

How these claims should be reconciled may depend on the conventions of a particular domain. For example, a domain expert may want the graph to prioritize a particular source, retain multiple claims, or defer a decision when the evidence is uncertain. We introduce an intermediary policy layer that allows domain experts to specify their policies for graph reconciliation and requirements for lifecycle handling in natural language. We evaluate a policy-conditioned LLM and a policy-free LLM to measure the effect of this policy layer on graph-alignment accuracy using controlled claim pairs whose observations may also have expiration times.

This work makes four contributions. First, we introduce a natural-language policy layer that separates epistemic conflict classification from domain-specific graph resolution. Second, we model observation expiration as an auditable lifecycle transition processed in temporal order. Third, we construct a controlled vaccine-guidance benchmark of 75 observations, containing 36 claim pairs, and 11 expiration events. Finally, we distinguish policy-action alignment from policy-independent graph correctness, revealing that improved adherence to resolution policies does not necessarily produce a more accurate operational graph.

II. Related Work

A. Knowledge Graph Construction from Text

Recently, there has been substantial progress in knowledge graph generation from plain-text descriptions. One example is KG-Gen, a framework that extracts entities and relations as triples from plain text. It clusters similar candidates and uses a large language model to consolidate entities that are near-identical, mapping them to one canonical entity name while preserving the other variations as aliases. The intermediary knowledge graphs are then aggregated into one final graph [8].

While this approach is promising, KG-Gen does not explicitly handle temporal evidence or decide which relationship should remain operational when temporal claims conflict. When it ingests seemingly contradictory temporal triples, those relationships may be accumulated in the final graph instead of being reconciled. Our work uses KG-Gen as the underlying graph-construction library and adds a separate layer for temporal reconciliation.

B. Conflict Resolution

Existing KG fusion approaches commonly rely on automated aggregation and source-reliability heuristics, including voting, probabilistic truth discovery, and recency-based ranking, albeit unreliable as models struggle to justify selections for adaptive decision-making [3]. Although domain-aware and expert-in-the-loop methods exist, domain experts typically have limited ability to specify relation-specific reconciliation policies or override the fusion model's decision procedure [6]. Manual reconciliation is time-consuming, while selecting the claim supported by the largest number of sources may also be unreliable. This is especially true when sources are biased or copy information from one another [5].

Although these methods provide general strategies for resolving conflicting information, they provide limited support for domain experts to specify their own relation-specific and domain-specific policies. Our policy layer instead allows domain experts to state how different types of discrepancies should be resolved using natural-language instructions.

C. Temporal Knowledge Graphs

This concern is partially addressed by ZepAI's Graphiti [9]. Temporal evidence can be inserted into knowledge graphs under its framework. However, competing observations are generally handled through temporal invalidation or supersession, where newer evidence replaces a previously active observation.

Not every discrepancy should be treated as temporal supersession. Two claims may coexist, describe different levels of granularity, represent a correction, or remain uncertain. Our approach allows these cases to be handled differently according to the domain policy while preserving prior observations and their provenance for future audits.

III. Knowledge Graph Across Industry

Knowledge graphs are widely applicable in industry, often used in fields like e-commerce, specifically warehouse logistics for inventory tracking [7]. This structure has also been used to track and apply reasoning on the states of clinical recommendations [4, 12]. Claims in such industries are not timeless and real-world facts may also have a validity period themselves. When a new update arrives, practical knowledge graphs should support time-aware reasoning, and learn when to interpolate or replace historical facts [1]. Edge or node replacement operations should be in place when there's a contradiction between the data arriving, and the existing relationships in the knowledge graph. Past works across different disciplines have highlighted the importance of using a systemic approach to resolve conflicting observations or claims [4]. Each accepted source must also provide clear reasoning and justification for why it was selected over competing sources.

IV. Method Overview

We propose a design method that allows end-users, such as domain experts, to provide domain-specific policies for resolving discrepancies between competing claims. As a knowledge graph receives temporal evidence, inconsistencies may arise between new and existing claims. KG-Gen extracts entities and subject–predicate–object relations from the incoming evidence, after which our experimental temporal-ledger wrapper compares each new relation against related active facts already stored in the ledger. These ledger candidates are ranked by matching priority: an exact subject–object pair, a shared subject, and then a shared object. We selected KG-Gen over legacy extraction systems such as OpenIE because of its convenient Python setup and its ability to produce cleaner entities, whereas OpenIE can generate overly long or incoherent nodes [8, 10]. An LLM then classifies the relationship between the new evidence and each retrieved candidate using one of six epistemic labels: `duplicate`, `coexists`, `state_transition`, `correction`, `contradiction`, or `uncertain` [11]. This classification determines whether the claims express the same fact, can remain simultaneously active, represent successive states, correct an earlier assertion, conflict within the same scope and time interval, or cannot be reliably distinguished from the available evidence. Independently of the epistemic label, each comparison is assigned a semantic issue type, such as logical contradiction, temporal change, or granularity mismatch. The classifier also records relevant discrepancy metadata, including source provenance, source reliability, recency, temporal validity, negation, and numerical disagreement. The output per graph state also includes a justification on why the specific claim was selected out of all competing claims.

For each piece of temporal evidence description, the observation contains metadata with source provenance for grounded bookkeeping and becomes useful if domain experts want to manually intercept and resolve a state transition. For each domain, the end-user may provide natural-language evidence-resolution policies. Once produced, the first-pass epistemic classification is immutable: the second pass may determine how the evidence affects the current graph, but it cannot reclassify the relationship. The resolver returns one of five actions: `accept_old`, `accept_new`, `retain_both`, `defer`, or `qualify`. If the enabled JSON policy contains an instruction for the classified relationship, the second pass applies that domain-specific guidance. An enabled policy with an empty instruction map invokes default resolution for every relationship. When no policy is enabled, a policy-free generic resolver runs only if explicitly configured. Otherwise, no second resolution pass occurs. The resolver does not automatically prefer the newer observation, as it may select either claim, retain both, defer the decision, or produce a qualified assertion. These actions modify the current graph projection while preserving the underlying evidence.

A. Observations with Expiration

The metadata also offer an optional expiration date of each observation. When an observation reaches its expiration time, its transition to the `expired` state is deterministic. If a lifecycle policy is configured, the LLM applies it only to describe domain-specific review or follow-up consequences. Pending events are maintained in a min-heap, with the event with the earliest expiration at the top. Before ingesting a new observation, the ledger processes every expiration whose timestamp is less than or equal to the new observation's timestamp, and the system resolves the pending observation first before handling the new observation, subsequently popping it from the heap. After its removal, this observation is saved as a recorded entity for future audits, rather than being discarded. Natural-language instructions for scheduled lifecycle events are stored in the same JSON policy under a separate `lifecycle_instructions` section.

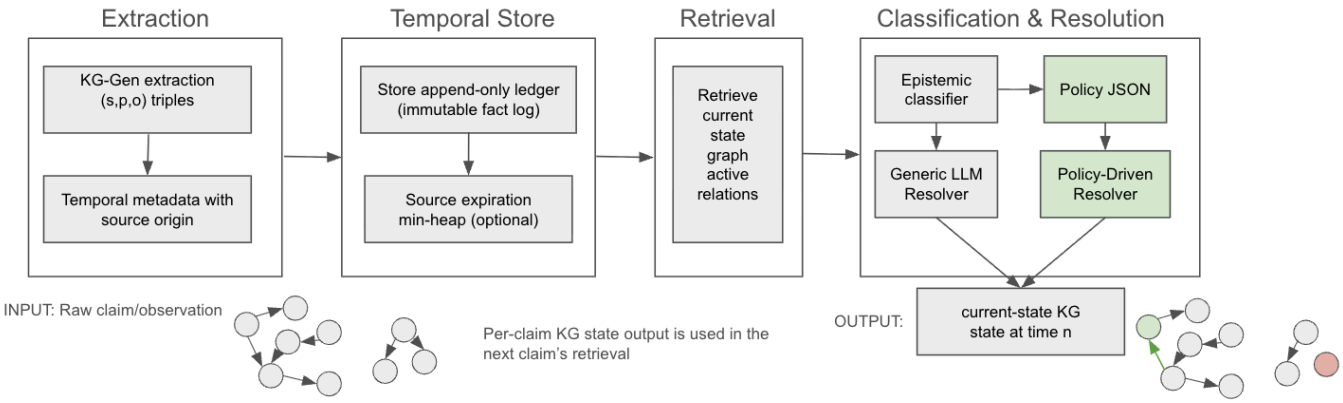

Fig. 1. Pipeline for ingesting a new claim.

V. Experimental Setup

We ran our experiments across different scenarios covering diverse fields and industries that record updates as knowledge graphs. To create contradictions and conflicts, we synthetically generated temporal pairs of field observations with descriptions that introduce nuanced differences. Each pair was synthesized using OpenAI's GPT-5.6 Sol model and interwoven with other pairs so that the observations do not appear side by side in a state transition.

These nuances represent types of potential discrepancies that may eventually be resolved by our epistemic keyword policy resolver. For example, event B may include exceptions that event A did not capture. In other cases, events A and B may be mutually exclusive, resulting in a logical contradiction that must be resolved according to the specified conflict-resolution policy.

For each scenario benchmark, we sampled sources with guidelines on how that specific industry or field handles conflicting information. In the vaccine benchmark, we provided context to the LLM on government sources from the Centers for Disease Control and Prevention for contraindications conventions, dosage reporting, etc. From these guidelines, the LLM follows a policy template where select epistemic keywords have their own resolution policy laid out in natural language.

Given the same domain-specific source guidelines and each pair of discrepant observations, GPT-5.6 Sol generates the ground truth by determining which observation the knowledge graph should resolve to based on the original source guidelines. We used another LLM to run the evaluation, recording metrics on how accurate the knowledge graph conflict resolutions align with the field-specific guidelines. Each policy-driven experiment run is compared with a dry run equivalent experiment where no policy resolver JSON exists, which means the LLM generically handles pairs of mismatching temporal claims. As shown in Fig. 1, our pipeline may invoke up to three LLM requests per claim: once during extraction for breaking down natural language claims into `(subject, predicate,object)` triples, one to classify the epistemic relationship, and an optional third to apply a resolution instruction when the classification matches an active entry in the policy file. Graph construction itself does not use an LLM.

VI. Experimental Result

In the vaccine scenario benchmark, we ran the experiment on 75 observations: 72 observations forming 36 discrepant claim pairs and three standalone lifecycle observations. Across the dataset, 11 observations had validity deadlines that triggered scheduled expiration events. We built the knowledge graph using the KG-Gen library coupled with Qwen3.6-35B-A3B with reasoning as the evaluated LLM. For each claim pair, the gold resolution represents the reference outcome established during benchmark construction from the domain-specific source guidelines. It specifies the expected epistemic relationship, such as contradiction, coexistence, or state transition, as well as the expected resolution action, such as accepting the new claim, retaining both claims, qualifying them, or deferring the decision. Conditional policy action accuracy measures the ratio of correct resolution actions among policy-governed cases that the evaluated LLM first routed to the correct epistemic relationship. For lifecycle events, the gold resolution specifies the expected expiration action, whether the historical claim should be preserved, and whether review is required. Lifecycle resolution accuracy is the proportion of the 11 expiration events for which all three fields matched the gold resolution.

As shown in Table I, both runs achieve the same aggregate issue-type accuracy, suggesting that the policy primarily affects downstream conflict resolution rather than initial conflict classification. This equality does not necessarily mean that both runs classify every individual case identically. Policy-grounded decision accuracy evaluates whether the case is routed to the correct conflict type, the resolution action matches the gold action, the resulting active facts match the expected graph state, and the required policy safeguards are

TABLE I
POLICY ALIGNMENT METRICS FOR THE VACCINE SCENARIO BENCHMARK. DELTA POLICY LLM MINUS THE GENERIC LLM.

Metric	Policy LLM	Generic LLM	Delta
Policy-grounded decision accuracy	39.3%	25.0%	+14.3 pp
Policy routing coverage	60.7%	64.3%	-3.6 pp
Conditional policy action accuracy	64.7%	38.9%	+25.8 pp
Issue-type accuracy	72.2%	72.2%	0.0 pp
Resolution accuracy	30.6%	19.4%	+11.2 pp
Lifecycle resolution accuracy	72.7%	45.5%	+27.2 pp

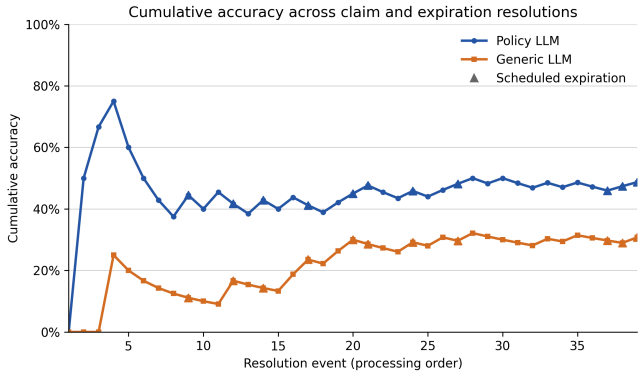

Fig. 2. Cumulative policy-grounded decision accuracy over the sequence of claim-resolution and expiration events.

satisfied. Conditional policy action accuracy isolates the performance of the resolver after a case has already been routed to the correct conflict type. Resolution accuracy is the broadest of the three metrics: it measures only whether the predicted resolution action matches the gold action across all claim pairs, without requiring correct policy routing or verifying the resulting graph state. The policy-grounded decision accuracy rate can be represented with this equation:

$$\frac{1}{|\mathcal{D}\text{pol}|} \sum_{i \in \mathcal{D}_{\text{pol}}} \mathbf{1}\left[\hat{r}_i = r_i^* \wedge \hat{a}_i = a_i^* \wedge \hat{G}_i = G_i^*\right]$$

where $\mathcal{D}_{\text{pol}}$ is the set of policy-governed benchmark cases. For a case to be counted as correct, the predicted conflict type $\hat{r}_i$, resolution action $\hat{a}_i$, and active graph state $\hat{G}_i$ must match their respective gold values $r_i^\star$, $a_i^\star$, and $G_i^\star$. The term C_i indicates whether the required policy safeguards are satisfied. Policy-grounded decision accuracy is therefore the proportion of policy-governed cases for which the complete resolution process is correct across the temporal span.

In Figure 2, we pooled temporal and non-temporal claim-resolution pairs into a single line graph to compare how policy-grounded decision accuracy evolves as the same sequence of conflicting claims is presented over time. This graph alone does not establish that accuracy generally degrades over time; it presents only an observation from the vaccine scenario. The results illustrate how synthetically generated knowledge graphs may deviate from the ground truth, particularly when their connected components have high degrees. For this particular synthetic dataset on vaccines, Policy LLM produced a cumulative of 48.7%, while generic LLM produced a cumulative accuracy of 30.8%, a 58% improvement overall.

VII. LIMITATIONS AND CONCLUSION

Our experiments were not verified by industry experts for each scenario. While the scenario use cases and their applicability are supported by past work [4, 7], each scenario was generated with an LLM to evaluate the effectiveness of an additional layer that classifies the type of conflict and executes a resolution policy. This policy selects the most suitable claim according to how conflicts are handled within the specific domain. The target edges are also synthetic and serve as a baseline for highlighting misalignment when conflicts are resolved using a policy from a different domain or no policy at all. Carrying forward the limitations of KG-Gen, our experiment does not evaluate dense corpora and follows KG-Gen's custom data schema, which differs in structure from traditional ontology languages and schema modeling frameworks such as the Web Ontology Language (OWL) and LinkML. The original MINE benchmark used by KG-Gen was itself relatively small, evaluating 100 articles of approximately 1,000 words each. In the vaccine benchmark, the connected components had low degrees of connectivity and were not tested on deeply nested knowledge graphs. Therefore, these results cannot be used to conclude that the approach is effective across graph structures with varying levels of complexity. Furthermore, we ran our experiments on only two scenarios, with one case study reported in this paper. More benchmarks across diverse scenarios are needed to determine whether our policy-layer approach is effective across additional disciplines.

REFERENCES

[1] Li Cai et al. *A Survey on Temporal Knowledge Graph: Representation Learning and Applications*. 2024. arXiv: 2403.04782 [cs.CL]. URL: https://arxiv.org/abs/2403.04782.

[2] Centers for Disease Control and Prevention. *Timing and Spacing of Immunobiologics*. Updated July 24, 2024; accessed September 17, 2026. July 2024. URL: https://www.cdc.gov/vaccines/hcp/imz-best-practices/timing-spacing-immunobiologics.html.

[3] Hejie Cui et al. *A Review on Knowledge Graphs for Healthcare: Resources, Applications, and Promises*. 2025. arXiv: 2306.04802 [cs.AI]. URL: https://arxiv.org/abs/2306.04802.

[4] Kristijonas Čyras and Tiago Oliveira. *Resolving Conflicts in Clinical Guidelines using Argumentation*. 2019. arXiv: 1902.07526 [cs.AI]. URL: https://arxiv.org/abs/1902.07526.

[5] Xin Luna Dong, Laure Berti-Equille, and Divesh Srivastava. "Data Fusion: Resolving Conflicts from Multiple Sources". In: *Web-Age Information Management*. Vol. 7923. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2013, pp. 64–76. DOI: 10.1007/978-3-642-38562-9_7. URL: https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/41657.pdf.

[6] Lucas Jarnac, Yoan Chabot, and Miguel Couceiro. *Uncertainty Management in the Construction of Knowledge Graphs: a Survey*. 2024. arXiv: 2405.16929 [cs.AI]. URL: https://arxiv.org/abs/2405.16929.

[7] Feng-Lin Li et al. "AliMeKG: Domain Knowledge Graph Construction and Application in E-commerce". In: *Proceedings of the 29th ACM International Conference on Information amp; Knowledge Management*. CIKM '20. ACM, Oct. 2020, pp. 2581–2588. DOI: 10.1145/3340531.3412685. URL: http://dx.doi.org/10.1145/3340531.3412685.

[8] Belinda Mo et al. *KGGen: Extracting Knowledge Graphs from Plain Text with Language Models*. 2025. arXiv: 2502.09956 [cs.CL]. URL: https://arxiv.org/abs/2502.09956.

[9] Preston Rasmussen et al. *Zep: A Temporal Knowledge Graph Architecture for Agent Memory*. 2025. arXiv: 2501.13956 [cs.CL]. URL: https://arxiv.org/abs/2501.13956.

[10] Philippe Remy. *Python wrapper for Stanford OpenIE*. https://github.com/philipperemy/Stanford-OpenIE-Python. 2020.

[11] James Thorne et al. *FEVER: a large-scale dataset for Fact Extraction and VERification*. 2018. arXiv: 1803.05355 [cs.CL]. URL: https://arxiv.org/abs/1803.05355.

[12] Jian Xu et al. *PubMed knowledge graph 2.0: Connecting papers, patents, and clinical trials in biomedical science*. 2025. DOI: 10.48550/arXiv.2410.07969. arXiv: 2410.07969 [cs.DL]. URL: https://arxiv.org/abs/2410.07969.