TRACE-OU-Bench: A Sequential Benchmark for Ontology Update Decisions from Financial Filings

Fang-Chun Yeh[1], Urjitkumar Patel[1] and Chinmay Gondhalekar[1]

[1]S&P Global, New York, United States Of America

Abstract

Ontology maintenance increasingly requires deciding not only what new schema element may be useful, but whether a concrete proposed update is warranted under the ontology state available at that time. Existing work has explored automated ontology-extension prediction and dynamic schema induction, yet realistic resources for evaluating proposal-level update decisions across evolving ontology snapshots remain limited. We introduce TRACE-OU-Bench, a sequential benchmark built from public SEC 10-Q filings. It contains 620 human-reviewed ontology-update candidates from 45 U.S. companies in five sectors over 2016–2026. Candidates are generated and reviewed chronologically: Train uses O_0 and builds O_1, Calibration uses O_1 and builds O_2, and Test uses O_2, so no split ever sees future schema state. Each record links source provenance and proposal-time ontology context to a proposed operation, a four-way human decision, structured corrections, and component-level labels. The benchmark exposes a sharp shift in the proposed-action distribution: NO_UPDATE proposals rise from 38.6% in Train to 96.7% in Test. Crucially, this does not make the decision trivial: only 44.7% of Test proposals are accepted as-is, so most NO_UPDATE proposals (the dominant action) are themselves incorrect. The central challenge is therefore detecting false NO_UPDATE decisions rather than ranking a balanced pool. Static acceptance baselines transfer poorly (Test AUROC 0.52), while evidence-correctness prediction remains more learnable (AUROC 0.74). We release TRACE-OU-Bench to support snapshot-aware, continual, and human-in-the-loop ontology-evolution research.

Keywords

ontology evolution, knowledge graphs, benchmark, financial filings, human adjudication, large language models

CEUR-WS.org/Vol-4316/pd-short-8.pdf

1. Introduction

Ontologies used in continuously updated knowledge graphs (KGs) are not static. New documents can introduce missing relations, reveal that a relation signature is too narrow, or show that an apparently novel proposal is already covered by the current schema. This creates a deployment problem that is easy to miss in conventional evaluation: the correct decision for a proposal depends on the ontology state available when that proposal was generated.

Consider an illustrative example. A filing states that a company depends on an external supplier. Under an early snapshot O_0 with no suitable dependency relation, an ADD_RELATION proposal may be justified. After a reviewed update introduces a broader dependency relation in O_1, a semantically similar later proposal may correctly become NO_UPDATE. A random split can mix these incompatible decision contexts even though they would not coexist at deployment time.

This motivates the central thesis of this paper: *ontology-update decisions should be evaluated against the ontology state that actually existed at proposal time.* Otherwise, evaluation can obscure the distribution shift induced jointly by later documents, an evolving schema, and a fixed proposal-generation pipeline.

Prior work makes automated ontology maintenance increasingly plausible, from classical evolution-management frameworks [1, 2], data-driven ontology learning and extension prediction [3, 4], and factual KG updating [5] to LLM-based ontology learning [6], dynamic schema induction [7], and LLM-generated KG judging [8]. What remains less developed is proposal-level evaluation that preserves the active ontology snapshot and regenerates later candidates only after earlier reviewed updates.

We introduce TRACE-OU-Bench, a sequential benchmark built from public SEC 10-Q filings. The

IJCKG 2026 Posters, Demos and Workshops, November 19–21, 2026, Bangkok, Thailand

jessie.yeh@spglobal.com (F. Yeh); urjitkumar.patel@spglobal.com (U. Patel); chinmay.gondhalekar@spglobal.com (C. Gondhalekar)

© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

published 2026-11-06

Table 1
Task assumptions in selected related work. "Proposal" means a concrete candidate whose accept/correct/reject decision is evaluated; Gov. = governance.

Work	Update target	Proposal	Active O_t	Regen.	Evidence	Correction
Pesquita–Couto [4]	ontology area	No	Hist.	No	Ext. feats.	No
TICO [10]	classes/properties	Action	Yes	Stream	Instances	No
Tang et al. [5]	KG facts	No	KG	Continual	Text	No
DIAL-KG [7]	facts/schema	Partial	Yes	Yes	Text	Gov.
KGCL [11]	change repr.	Change	Yes	N/A	N/A	Curator
OCP2KG [12]	mappings	Change	Yes	N/A	N/A	No
GraphJudge [8]	KG facts	Yes	No	No	Text	Judge
TRACE-OU-Bench	relation schema	Yes	Yes	Yes	Text	Yes

benchmark does not claim that a particular classifier solves ontology updating. Instead, it provides the decision contexts needed to test future methods fairly. Its contributions are: (1) a chronological $O_0 \to O_1 \to O_2$ construction protocol; (2) 620 self-contained human-reviewed proposal records with four-way decisions, structured corrections, component labels, and error types; and (3) a characterization of annotation reliability, prefilter coverage, candidate shift, and frozen baselines with document-grouped uncertainty estimates.

This design supports three research questions: How well do decision models trained on an earlier ontology state transfer to later documents and updated snapshots? Can models distinguish correct NO_UPDATE proposals from missed update opportunities? Can component-level supervision and structured corrections support diagnosis and repair when accept-as-is prediction is unreliable? The benchmark provides evaluation contexts for these questions; the reference baselines characterize their difficulty.

2. Related Work and Task Positioning

Prior ontology-evolution research studies schema-change detection, management, and extension prediction [2, 4], but generally does not evaluate document-grounded proposals against the ontology state available at proposal time.

Recent LLM-based work has expanded automated ontology and knowledge-graph construction. LLMs4OL formulates ontology-learning tasks such as term typing, taxonomy discovery, and non-taxonomic relation extraction [6], while DIAL-KG studies continual KG construction with governance adjudication and dynamic schema induction [7]. GraphJudge further examines whether LLM-generated KG content can be reliably judged [8]. TRACE-OU-Bench complements these efforts by fixing the proposal-generation setting and pairing human-reviewed proposals with their active ontology snapshot, source provenance, structured corrections, and component-level correctness labels. Together, these artifacts support evaluation of whether a particular proposal is justified under its historical schema context and which fields require correction.

Temporal evaluation is important because training and deployment conditions can diverge over time, and temporal misalignment has been shown to degrade predictive performance [9]. In ontology updating, the problem is stronger than ordinary temporal drift because the schema itself changes: a proposal that is warranted under an earlier ontology may become redundant after reviewed updates are incorporated. TRACE-OU-Bench therefore regenerates later candidates only after earlier human-gold updates have produced the next ontology snapshot. Table 1 summarizes selected task-level distinctions. The contribution of TRACE-OU-Bench is the joint evaluation setting: proposals are judged against their proposal-time ontology state, and later candidates are regenerated only after earlier reviewed updates. Four-way adjudication and structured corrections additionally distinguish proposals that are acceptable as written from useful ideas that require repair. This supports studying transfer across evolving snapshots and correction needs within individual proposals.

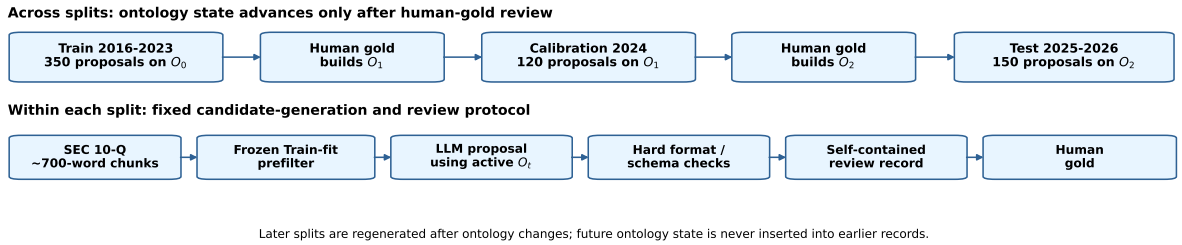


Figure 1: Two levels of the benchmark. Across splits, human-gold updates advance the ontology state. Within a split, every proposal is generated and reviewed against the same frozen active snapshot.

3. Task Definition

For proposal time t, let x_t denote the source filing context, O_t the active ontology snapshot, and p_t an ontology-update proposal. The primary adjudication task is

$$y_t = f(x_t, O_t, p_t), \qquad y_t \in \{\textsc{Accept}, \textsc{Modify}, \textsc{Reject}, \textsc{Insufficient}\}.$$

Successful reasoning may require verifying that evidence is present and supports the proposal, checking whether O_t already covers the semantics, validating operation choice and target relation, checking domain/range compatibility, and deciding whether the text is sufficiently specific for a directional schema update.

Three variables must be kept separate. **Proposal action** is what the generator suggests (e.g., AD D_RELATION or NO_UPDATE). **Adjudication decision** is the human judgment of that proposal. **Effective schema outcome** is the operation actually applied after review: ACCEPT uses the proposal as written; MODIFY uses corrected fields; REJECT and INSUFFICIENT apply no schema operation. An accepted NO_UPDATE is therefore a correct proposal that intentionally changes nothing. This explains why the percentage of NO_UPDATE proposals and the accept-as-is rate are not complements.

4. Benchmark Construction

4.1. Documents, Seed Ontology, and Records

The source corpus consists of public SEC 10-Q filings from 45 U.S. companies in Banking, Energy, Healthcare, Retail and Consumer, and Technology. Reviewed material spans 2016–2026 and focuses on Management's Discussion and Analysis and Risk Factors. Sections are split into approximately 700-word chunks with 100-word overlap. The seed snapshot O_0 contains a fixed hierarchy of 45 entity classes and 20 active relation schemas; relation schemas evolve while the class hierarchy remains fixed.

The original experimental records contain filing text and generator evidence quotes alongside filing metadata, the active snapshot and readable relation inventory, proposal fields, rationale and confidence, human decisions and corrections, component labels, and repeat/audit metadata. The public release retains structured proposal and annotation fields while replacing source chunks and raw evidence quotations with source URLs, deterministic identifiers, offsets, alignment status, and cryptographic hashes. Source-matched passages detected in free-text fields are redacted as documented in the release. Preserving the active snapshot keeps each adjudication conditional on the relation inventory available at proposal time.

4.2. Frozen Train-Fit Prefilter and Candidate Generator

The prefilter is fitted only on Train and frozen thereafter, and the same fitted pipeline is applied unchanged to Calibration and Test. It combines lexical, semantic, and section-level signals to select candidate source chunks.

Table 2
Sequential split summary. "Accept-as-is" is the adjudication target; NO_UPDATE is the generator's proposed action.

Split	Years	N	Accept-as-is	NO_UPDATE	Repeats
Train	2016–23	350	27.7%	38.6%	35
Calibration	2024	120	47.5%	94.2%	12
Test	2025–26	150	44.7%	96.7%	15

Selected chunks are passed to a fixed local LLM generator conditioned on the full active relation inventory of the corresponding snapshot O_t. The generator proposes relation-schema updates or NO_UPDATE when no change is justified. Generated outputs undergo hard format and schema-validity checks; exact duplicates are collapsed, while malformed or schema-invalid proposals are excluded. Evidence/source alignment is preserved as an observable signal rather than silently repaired.

From the resulting valid candidate pool, reviewed proposals are sampled in round-robin strata defined by action and industry, yielding 350 Train, 120 Calibration, and 150 Test candidates.

4.3. Sequential Snapshot Construction

Snapshot update algorithm. The procedure is deterministic: (1) freeze O_t and generate the split; (2) validate every effective ACCEPT/MODIFY operation; (3) ACCEPT keeps the proposal and MODIFY substitutes only corrected fields, with blanks inheriting the original; (4) REJECT, INSUFFICIENT, and effective NO_UPDATE apply no change; (5) apply effective updates ordered by filing date and candidate ID, collapsing duplicate new signatures within the split as corroborating support while conflicting parents or invalid targets fail closed; and (6) save O_{t+1} before regenerating the next split. Train gold builds O_1, Calibration gold builds O_2, and Test does not build O_3.

5. Human Annotation and Quality Control

5.1. Four-Way Decisions and Structured Corrections

During annotation, a single primary human annotator reviewed the complete experimental rows containing source text, the original generator evidence quote when supplied, active relation inventory, proposal, target context, LLM rationale, confidence, filing date, and split metadata. The benchmark therefore measures human-adjudicated proposal quality under a written protocol; it does not claim inter-annotator consensus. The decision labels are: Accept (safe as written), Modify (useful core idea but at least one structured field must change), Reject (unwarranted, redundant, or structurally wrong), and Insufficient Evidence (text/snapshot context cannot support a grounded directional decision). For MODIFY, only changed fields are filled and all other corrected fields inherit the original proposal.

Five component labels are retained: evidence_correct, target_correct, operation_correct, temporal_c orrect, and downstream_safe. The binary task label gold_accept equals one only for ACCEPT. The Train decision distribution is 97 ACCEPT, 153 MODIFY, 96 REJECT, and 4 INSUFFICIENT; thus 43.7% of Train proposals preserve a useful idea but require correction, information that binary labeling would discard.

5.2. Blinded Repeats and Prefilter Audit

Ten percent of reviewed candidates were duplicated as blinded repeat rows and interleaved in the same shuffled annotation file with candidate IDs hidden. The repeat set was random within each split rather than separately stratified. Because one primary annotator reviewed all rows, the reported Cohen's κ values measure *intra-annotator repeat consistency*. Final-decision Cohen's κ is 0.957 on Train, 0.844 on Calibration, and 1.000 on Test; detailed component-level agreement is included in the release.

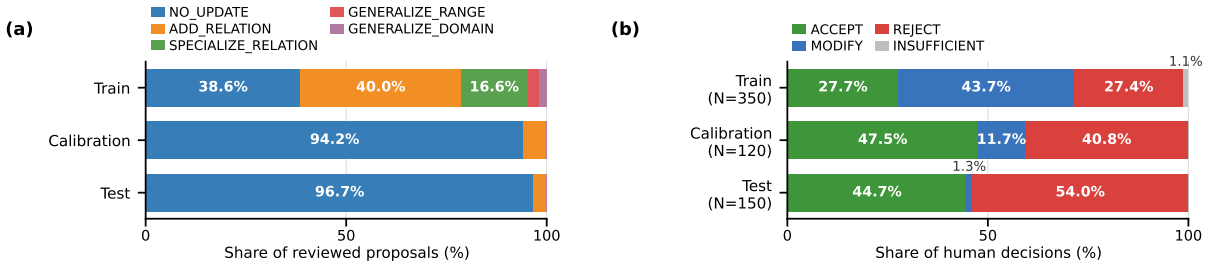


Figure 2: Benchmark characteristics across chronological splits. (a) Generated action composition by proposal-time snapshot. Later splits are dominated by NO_UPDATE; this is an observed composite shift rather than a claim about a single cause. (b) Human adjudication across sequential splits. Test remains divided between ACCEPT and REJECT even though proposed actions are mostly NO_UPDATE; MODIFY falls from 43.7% in Train to 1.3% in Test.

The same file includes a prefilter audit sampled 40/30/30% from selected, near-miss, and low-score strata; weights are the stratum population/sample ratio. Annotators mark whether a chunk contains an update opportunity and whether generated candidates give FULL, PARTIAL, or NONE coverage. Weighted recall is 0.888 on Train and 0.911 on Calibration. Test recall is undefined because 0/75 audited chunks are positive. The audit characterizes missed opportunities around the pipeline; it does not prove exhaustive enumeration of all possible updates.

5.3. Proposal Error Types

The release also retains fine-grained primary_error_type labels that distinguish generator failures from adjudication errors, including evidence mismatch, wrong domain/range, redundant specialization, unnecessary generalization, existing-relation coverage, and false NO_UPDATE.

6. Benchmark Tasks and Evaluation Protocol

TRACE-OU-Bench supports four tasks: T1 four-way adjudication (Macro-F1, per-class F1); T2 accept-as-is ranking/probability prediction (AUROC, AUPRC, Brier/NLL); T3 component correctness (component AUROC/AUPRC with positive/negative counts); and T4 structured correction (field accuracy, exact corrected signature, schema-validity checks).

The canonical protocol is chronological: fit learned decision models on Train, use Calibration only for model selection or probability calibration, and evaluate Test once. Allowed inputs are source context, proposal fields (including generator confidence when explicitly disclosed), active snapshot context, and non-gold metadata. Gold decisions, structured corrections, component labels, error labels, and private repeat mappings are prohibited as model inputs. Pretrained encoders or external models may be used but should be declared. Component metrics are computed only on rows with defined component gold labels. Randomly reshuffled evaluation may be reported as an ablation, but it is not the canonical benchmark because it mixes proposal-time schema states.

Evaluations using the public release should disclose the release version, input availability, and any substituted inputs or restricted candidate subsets (see Section 8).

7. Benchmark Characterization and Baselines

7.1. Observed Deployment-Like Shift

Figure 2(a) shows that the generated action distribution changes sharply across snapshots. Train contains 140 ADD_RELATION, 135 NO_UPDATE, 58 SPECIALIZE_RELATION, 10 GENERALIZE_RANGE, and 7 GENERALIZE_DOMAIN candidates. Calibration contains 113 NO_UPDATE and 7 ADD cases;

Test contains 145 and 5, respectively. We retain rather than rebalance this shift because later candidate scarcity is itself a deployment condition the benchmark is meant to expose.

The shift must therefore be interpreted carefully. It is consistent with an ontology that has accumulated useful relations, but alternative explanations include changes in filing language, proposal-generator behavior under a larger relation inventory, prefilter interactions, and action-mixture effects. The benchmark measures this combined longitudinal shift rather than claiming ontology maturity as the sole cause.

The change in proposed actions should not be confused with the distribution of human decisions. Although 96.7% of Test proposals use the NO_UPDATE action, only 44.7% of Test proposals are accepted as-is, while 54.0% are rejected and 1.3% require modification (Figure 2(b)). The Test split is therefore not a near-constant decision problem: the central challenge is distinguishing correct NO_UPDATE proposals from false NO_UPDATE proposals. The adjudication distribution also changes qualitatively across snapshots. MODIFY falls from 43.7% in Train to 11.7% in Calibration and 1.3% in Test, while REJECT rises from 27.4% to 54.0%. This suggests that later proposals are less frequently recoverable through structured correction, although the benchmark does not isolate whether this pattern is driven by ontology growth, temporal document differences, or generator behavior.

7.2. T2 Reference Baselines

We evaluate five reference methods for the binary accept-as-is task (T2). **Constant Calibration prior** assigns every candidate the Calibration-set ACCEPT prevalence of 0.475. **Train-only action frequency** assigns each candidate the empirical Train ACCEPT rate of its proposed action. **Raw LLM confidence** uses the proposal generator's self-reported confidence directly as the ACCEPT score. **Direct binary logistic regression** predicts gold_accept from the full feature set using a single Train-only logistic-regression model. Finally, the **Component-aware stack** first predicts evidence, target, operation, and temporal correctness and then combines these component probabilities in a second Train-only logistic-regression meta-model.

The learned baselines use Train-fitted proposal, evidence, ontology-snapshot, target, and categorical features. Filing year, text-length proxies, raw relation/type identifiers, counterexample-verifier features, and class weighting are excluded.

The Component-aware stack combines probabilistic component predictions in a second-stage logistic model. Train meta-features use document-grouped out-of-fold predictions to prevent in-sample leakage; downstream_safe is excluded because it duplicates gold_accept on Train.

Table 3 reports Test performance with 95% document-grouped bootstrap confidence intervals based on 2,000 replicates. Train-only action frequency achieves the highest AUROC, 0.517 [0.491, 0.544], with an interval that includes chance. Its limited discrimination is expected because all 145 Test NO_UPDATE proposals receive the same action-based score, although their human decisions differ.

The learned models also transfer poorly: Direct binary logistic regression reaches AUROC 0.363, and the Component-aware stack reaches 0.324 with AUPRC 0.372. They are trained on an earlier mixture with substantially more additions and MODIFY decisions, whereas later evaluation depends heavily on distinguishing correct from false NO_UPDATE proposals. Feature–label associations learned from the earlier mixture may therefore be poorly aligned with later adjudication requirements. Changes in filing language and generator behavior under an expanded relation inventory may further contribute. These are plausible explanations consistent with the observed shift; the present construction does not isolate their individual effects.

Snapshot-context ablation. Removing active-ontology and target-related features from the Component-aware stack yields Test AUROC 0.318 [0.226, 0.417] and AUPRC 0.396, versus 0.324 [0.229, 0.421] and 0.372 for the full model. The overlapping AUROC intervals provide no evidence that the current snapshot features improve chronological ACCEPT prediction. This does not imply that ontology state is irrelevant; rather, effectively representing it remains an open challenge exposed by TRACE-OU-Bench.

Table 3
T2 Test baselines. AUROC intervals are 95% document-grouped bootstrap CIs (2,000 replicates); AUPRC point estimates are shown, with matched CIs reported in the release artifacts. Test ACCEPT prevalence is 0.447.

Method	AUROC [95% CI]	AUPRC
Constant Calibration prior	0.500 [0.500, 0.500]	0.447
Train-only action frequency	0.517 [0.491, 0.544]	0.455
Raw LLM confidence	0.352 [0.292, 0.410]	0.389
Direct binary logistic regression	0.363 [0.269, 0.460]	0.383
Component-aware stack	0.324 [0.229, 0.421]	0.372

For T3, evidence correctness is the only component with sufficient class balance for meaningful interpretation, achieving Test AUROC 0.743 with 71 positives and 79 negatives. This contrasts with the weak accept-as-is results: identifying evidence correctness alone does not establish that the operation, target, or schema-level decision is warranted under the active snapshot. The component task therefore offers a useful diagnostic direction, while successful composition into proposal acceptance remains an open problem. Target, operation, and temporal correctness each contain only three to four negative cases, so we withhold their AUROC point estimates as not informative.

Probability diagnostics are similarly cautionary. Because raw confidence is not necessarily calibrated in either generic neural classifiers [13] or knowledge graph models [14, 15], we treat calibration as a diagnostic rather than evidence of successful automation. Calibration collapses near the Calibration prior, and selective risk policies automate no Test cases, yielding no review savings. We therefore treat these results as diagnostics of benchmark difficulty rather than performance contributions.

8. Validity, Limitations, and Availability

Limitations and validity. TRACE-OU-Bench is a controlled longitudinal testbed rather than a universal sample of ontology-evolution settings. First, all records were reviewed by one primary annotator; blinded repeats therefore measure intra-annotator consistency rather than independent inter-annotator agreement. Second, the benchmark currently covers one document family (SEC 10-Q filings), one application domain, a fixed entity-class hierarchy, and 620 reviewed proposals, while several Test component labels are highly imbalanced. Third, ontology updates occur between chronological splits rather than after every individual filing, so the setting approximates periodic ontology maintenance rather than fully online evolution. Finally, the observed longitudinal shift reflects the combined effects of temporal changes in filing language, the frozen Train-fitted prefilter, the fixed proposal generator, the evolving ontology state, and changing action mixtures. We therefore do not attribute poor chronological transfer to ontology growth alone. Likewise, ACCEPT denotes that a proposal can be applied as written under the active snapshot O_t, not that the proposed relation is universally optimal. The reported results should therefore be interpreted primarily as a characterization of proposal-level transfer difficulty under a sequential ontology-update setting.

Reproducibility and public availability. The public release contains the 620 reviewed candidate records, chronological split assignments, proposal-time snapshot identifiers and relation inventories, structured corrections, component labels, ontology snapshots, prefilter-audit records, and blinded-repeat mappings. To avoid redistributing filing excerpts, it omits source chunk text and raw generator evidence quotations, including quotations nested in audit summaries. Detected source-text runs in rationale and annotation notes are redacted and documented.

SEC accession metadata, source URLs, deterministic chunk identifiers, offsets, hashes, and reconstruction and validation scripts support recovery of filing context. Validation of a user-performed reconstruction confirmed all 862 released source chunks against the manifest, including joins to all 620 candidate records and 250 audit records. These checks establish identity with the frozen source chunks; semantic section boundaries and annotation quality require separate assessment.

Among the 620 candidate records, evidence alignment is exact for 291, token-exact for 9, non-

extractable for 152, and absent for 168. The 300 aligned source spans were validated. Exact alignment permits recovery of the original quote; token-exact alignment recovers a corresponding source span without guaranteeing the original quote string. Non-extractable quotes remain unavailable. Alignment status describes textual correspondence and is separate from semantic correctness.

The reported baselines were computed from the original experimental records. Reconstruction of source chunks does not guarantee recovery of every original textual model input, including unavailable quotes and redacted free-text passages. The current data package includes reconstruction and validation code; complete baseline training and evaluation implementations are not bundled. Experiments using regenerated proposals or substituted inputs should report the resulting input and candidate differences. The repository is available at: https://huggingface.co/datasets/TRACE-OU/TRACE-OU-Bench.

9. Discussion and Conclusion

TRACE-OU-Bench evaluates ontology-update proposals against the ontology state available when they are generated. Its sequential design, correction-rich labels, and strong action shift expose substantial changes in proposal populations across snapshots, while simple baselines show weak transfer from earlier states.

The benchmark provides a controlled setting for studying snapshot-aware reasoning, continual adaptation, structured correction, and human-review routing. The same sequential protocol could be adapted to biomedical literature or dated regulatory documents by specifying a domain ontology, ordering incoming documents chronologically, reviewing proposals against a frozen active snapshot, and regenerating later candidates after reviewed updates. Adaptation would require domain-specific operations, constraints, and annotation guidance. For ontologies whose classes, taxonomies, or axioms also evolve, the current relation-schema operations and snapshot-validation rules would need extension. Cross-domain performance remains to be evaluated. Future work should compare chronological and random splits and test additional proposal generators to better isolate ontology-state effects from other temporal and pipeline shifts.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT for grammar and spelling checking, paraphrasing, and improving writing style. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] L. Stojanovic, A. Maedche, B. Motik, N. Stojanovic, User-driven ontology evolution management, in: Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web, volume 2473 of *Lecture Notes in Computer Science*, Springer, 2002, pp. 285–300. doi:10.1007/3-540-45810-7_27.

[2] F. Zablith, G. Antoniou, M. d'Aquin, G. Flouris, H. Kondylakis, E. Motta, D. Plexousakis, M. Sabou, Ontology evolution: A process-centric survey, The Knowledge Engineering Review 30 (2015) 45–75. doi:10.1017/S0269888913000349.

[3] P. Cimiano, J. Völker, Text2onto: A framework for ontology learning and data-driven change discovery, in: Natural Language Processing and Information Systems, volume 3513 of *Lecture Notes in Computer Science*, Springer, 2005, pp. 227–238. doi:10.1007/11428817_21.

[4] C. Pesquita, F. M. Couto, Predicting the extension of biomedical ontologies, PLoS Computational Biology 8 (2012) e1002630. doi:10.1371/journal.pcbi.1002630.

[5] J. Tang, Y. Feng, D. Zhao, Learning to update knowledge graphs by reading news, in: Proceedings of EMNLP-IJCNLP, Association for Computational Linguistics, 2019, pp. 2632–2641. doi:10.18653/v1/D19-1265.

[6] H. Babaei Giglou, J. D'Souza, S. Auer, LLMs4OL: Large language models for ontology learning, in: The Semantic Web – ISWC 2023, volume 14265 of *Lecture Notes in Computer Science*, Springer, 2023, pp. 408–427. doi:10.1007/978-3-031-47240-4_22.

[7] W. Bao, Y. Wang, R. Gao, F. Leng, Y. Bao, G. Yu, DIAL-KG: Schema-free incremental knowledge graph construction via dynamic schema induction and evolution-intent assessment, arXiv preprint arXiv:2603.20059 (2026).

[8] H. Huang, C. Chen, Z. Sheng, Y. Li, W. Zhang, Can LLMs be good graph judge for knowledge graph construction?, in: Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2025, pp. 10929–10948. doi:10.18653/v1/2025.emnlp-main.554.

[9] K. Luu, D. Khashabi, S. Gururangan, K. Mandyam, N. A. Smith, Time waits for no one! analysis and challenges of temporal misalignment, in: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, 2022, pp. 5944–5958. doi:10.18653/v1/2022.naacl-main.435.

[10] A. Canito, A. Nobre, J. Neves, J. Corchado, G. Marreiros, Using sensor data to detect time-constraints in ontology evolution, Integrated Computer-Aided Engineering 30 (2023). doi:10.3233/ICA-230703.

[11] H. Hegde, J. Vendetti, D. Goutte-Gattat, J. H. Caufield, J. B. Graybeal, N. L. Harris, N. Karam, C. Kindermann, N. Matentzoglu, J. A. Overton, M. A. Musen, C. J. Mungall, A change language for ontologies and knowledge graphs, Database 2025 (2025) baae133. doi:10.1093/database/baae133.

[12] D. Conde-Herreros, L. Stork, R. Pernisch, M. Poveda-Villalón, Ó. Corcho, D. Chaves-Fraga, Propagating ontology changes to declarative mappings in construction of knowledge graphs, in: Proceedings of the 5th International Workshop on Knowledge Graph Construction co-located with the 21st Extended Semantic Web Conference, volume 3718 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2024, pp. 1–16.

[13] C. Guo, G. Pleiss, Y. Sun, K. Q. Weinberger, On calibration of modern neural networks, in: Proceedings of the 34th International Conference on Machine Learning, volume 70 of *Proceedings of Machine Learning Research*, PMLR, 2017, pp. 1321–1330.

[14] P. Tabacof, L. Costabello, Probability calibration for knowledge graph embedding models, in: International Conference on Learning Representations, 2020.

[15] T. Safavi, D. Koutra, E. Meij, Evaluating the calibration of knowledge graph embeddings for trustworthy link prediction, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, 2020, pp. 8308–8321. doi:10.18653/v1/2020.emnlp-main.667.