Component-Aware Case Retrieval for Professional Ethics with Ontology-Constrained LLM Extraction

Christopher B. Rauch, J.D.[1] and Rosina O. Weber[1]

Drexel University, Philadelphia PA 19104, USA {cr625,rw37}@drexel.edu

Abstract. Professional ethics boards determine whether conduct conforms to established codes by comparing fact patterns against prior opinions and ethical code provisions, but identifying relevant precedents in a growing archive remains predominantly a manual process. Even when opinions are available in electronic form, text search is unlikely to surface complex ethical relationships between cases, and similarity scores based on embeddings provide no explanation for which aspects of two cases are related. We describe ProEthica, an applied CBR system that uses ontology-constrained LLM extraction to transform NSPE Board of Ethical Review cases into structured representations with nine components drawn from the computational ethics literature. The extraction produces named ontology entities for each component, embedded independently for retrieval, so similarity scores are traceable to the specific entities that produced them. Users can examine shared Principles, analogous Roles, differing Actions, or any combination of the nine components across retrieved precedents. Ground truth validation across 119 cases shows that structured methods recover cited precedents at 3 to 8 times the random baseline, outperforming section-based embedding.

Keywords: case-based reasoning · professional ethics · ontology-constrained extraction · component-aware retrieval · LLM knowledge acquisition

1 Introduction

Professional ethics boards determine whether professional conduct conforms to established codes by evaluating conduct against prior board opinions and code provisions. The National Society of Professional Engineers (NSPE) Board of Ethical Review (BER) has published over 650 advisory opinions since 1954, each applying the NSPE Code of Ethics to a specific fact pattern and citing prior board decisions as precedent. A board member reviewing a new case, whether based on a real situation or a hypothetical scenario posed to clarify a specific provision [23], identifies the relevant professional roles, determines which code provisions apply, evaluates what actions were taken and what alternatives existed, and compares the situation to prior cases with analogous features. This process follows the precedent-based adjudication model of common law reasoning [20], where structured comparison of prior decisions guides evaluation of new situations, and maps directly to the CBR cycle.

CBR has been applied to ethical analysis with structured case representations [5,21], but populating those representations from published board opinions remains a manual process. McLaren's [21] SIROCCO system demonstrated that case-based comparison of extensionally defined principles can predict which principles and precedents are relevant to new engineering ethics situations, but the system predates modern language models and requires hand-authored case representations (Section 2.1).

Computational ethics surveys [29,34] identify a broader gap between systems that apply abstract ethical principles and systems that engage with the structured, precedent-informed evaluation used in professional practice. Passing raw case text to an LLM can produce ethics analysis, but treating a case as an undifferentiated input discards the component-level structure that ethics boards rely on when evaluating professional actions [22].

ProEthica [25,26] is an applied CBR system for professional ethics analysis. The system uses ontology-constrained LLM extraction to decompose published NSPE Board of Ethical Review cases into nine typed components drawn from the computational ethics literature, each capturing a distinct aspect of professional ethical analysis (Section 3). A domain ontology provides formal definitions that constrain LLM output during extraction, and domain experts have the opportunity to review proposed ontology entities before they enter the shared vocabulary.

The paper makes two contributions. First, we formalize a nine-component case representation (Section 3.1) and populate it from published BER opinions through an ontology-constrained LLM extraction pipeline that produces named, typed entities grounded in a domain ontology. Similarity scores are traceable because each component embedding is generated directly from these entities. A high Principles similarity between two cases corresponds to specific Principle entities that users can inspect, compare across cases, and trace to the source text. The similarity network (Figure 2) provides component-level filters that identify where cases align or diverge. Second, we validate that this representation captures meaningful domain structure through a retrieval experiment comparing three embedding strategies under a shared weight configuration. Ground truth validation across 119 cases recovers cited precedents at 3 to 8 times the random baseline, with structured methods consistently outperforming section-based embedding. The near-perfect rank correlation between the two D-tuple aggregation strategies ($\rho = 0.991$) shows that aggregation choice has negligible effect on retrieval quality. A divergent-case analysis shows that the most discriminative component varies by case, a pattern that section-based embedding cannot capture.

Section 2 reviews related work. Section 3 describes the case representation and extraction pipeline. Section 4 presents the retrieval evaluation. Section 5 discusses implications and limitations. Section 6 concludes.

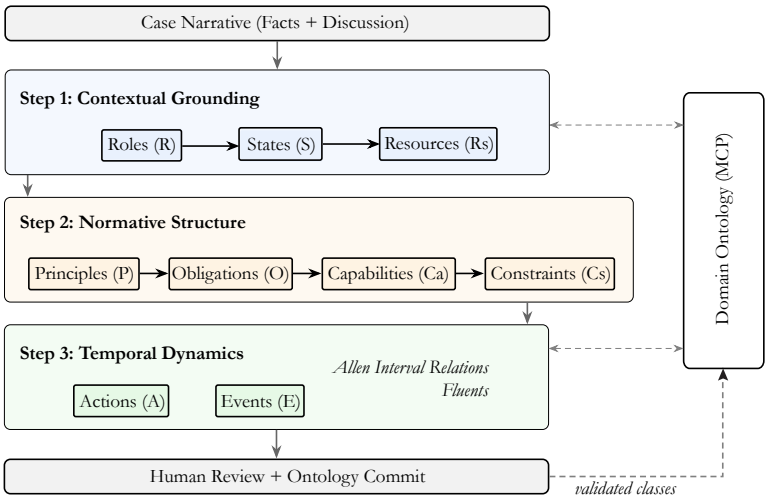

Fig. 1. Three-step extraction pipeline with dependency ordering. Solid arrows show extraction dependencies within and between steps. Dashed arrows show ontology queries (definitions injected before extraction) and the feedback loop where validated classes enter the shared ontology.

2 Background and Related Work

2.1 Professional Ethics and Precedent-Based Reasoning

Ashley and McLaren [5] developed Truth-Teller, a CBR system for practical ethics that compared pairs of cases using reasons, principles, and comparison rules to generate natural-language assessments of ethical similarities. McLaren [21] extended this work with SIROCCO, which defined ethical principles extensionally through case examples and encoded the associations between factual features and principles for retrieval based on normatively relevant characteristics. Both systems relied on manual knowledge engineering to construct case representations, a labor-intensive process that limited the size of their case bases.

2.2 CBR in Ethical and Legal Domains

Case-based approaches to ethical and legal reasoning share a common requirement. Cases must be represented at a level of granularity that supports meaningful comparison on normatively relevant features. Ashley's HYPO system [4] established the dimension-based reasoning tradition in legal CBR, where cases are indexed by legally significant dimensions rather than textual features alone. Bench-Capon and Sartor [7] extended this line of work by incorporating theories and values into case comparison, and their framework supports formal reasoning about the relative strength of competing arguments.

Recent work confirms that retrieval based on structured features produces better results than surface-level semantic matching. Van Woerkom et al. [33]

apply formal a fortiori reasoning to real-world legal datasets, demonstrating that case law contains measurable logical structure amenable to formal case comparison. Sun et al. [28] apply neuro-symbolic logic rules to legal case retrieval and show that structured features improve retrieval over surface-level semantic matching. Mumford et al. [22] find a complementary limitation: LLMs applied to European Court of Human Rights cases succeed at broad outcome prediction but fail at structured keyword classification, suggesting that surface-level processing cannot substitute for structured case analysis.

2.3 CBR and LLMs

The CBR-LLM Research Manifesto [6] identifies the knowledge acquisition bottleneck as one of the most prevalent challenges in CBR and a primary opportunity for LLM integration. Several recent systems address this opportunity. Ghazouani et al. [17] use LLMs to populate case bases by extracting entities from unstructured texts, while Brand et al. [11] demonstrate that LLMs can act as knowledge engineers to construct CBR knowledge containers by extrapolating from existing structured examples. Bergmann et al. [9] propose the EXAR architecture, which separates persistent knowledge units (long-term memory) from a per-task working context (working memory) to support incremental case acquisition across tasks.

A recurring concern across this literature is output fidelity. Wilkerson and Leake [31] document that LLM outputs include fabricated cases and altered details when applied to CBR tasks. Ge and Xu [16] implement the full 4R cycle for expense fraud detection and address the same concern through CBR-verified retention. The present work takes a complementary approach, constraining LLM output through ontology definitions (Section 3) with optional human review of proposed ontology entities.

2.4 CBR, RAG, and Structured Retrieval

Retrieval-augmented generation (RAG) has become the dominant approach for grounding LLM output in external knowledge, but standard implementations embed entire documents or passages as single vectors, discarding internal case structure. Wiratunga et al. [32] address this limitation in their CBR-RAG model for legal question answering, formalizing cases as structured tuples and applying dual embeddings optimized for attribute-to-attribute matching with a weighted retrieval function. Their work demonstrates that CBR-driven context selection outperforms generic document retrieval for domain-specific tasks. Lenz et al. [19] explore three approaches to LLM-based similarity assessment, finding that LLMs are most effective at configuring similarity measures rather than predicting similarity scores directly.

2.5 Theoretical Foundations

The nine-component D-tuple draws on several theoretical traditions. Oakley and Cocking [24] establish that professional roles generate distinct ethical requirements, and Dennis et al. [13] formalize this dependency, showing that abstract

principles must be translated into context-dependent obligations to be verifiable. These accounts produce the $R \rightarrow P \rightarrow O$ chain that forms the normative backbone of the representation. Legal doctrine independently reinforces this role-dependent structure. The professional negligence standard in the Restatement (Third) of Torts evaluates professionals against the standards of their specific profession rather than universal benchmarks [2], codifying into legal standards the same principle that role-based ethics establishes philosophically. Berreby et al. [10] demonstrate that ethical principles can be represented through modular, composable structures. This separates normative components (Principles, Obligations, Capabilities, Constraints) from contextual components (Roles, States, Resources). Allen's interval algebra [1], standardized through OWL-Time [12], provides the formal basis for temporal relations between Actions (A) and Events (E).

3 Case Representation and Ontology-Constrained Extraction

Professional ethics cases present a case representation challenge for CBR systems. Each case narrative incorporates professional roles, ethical standards, situational context, and temporal sequences of actions and events. McLaren [21] found that detailed case representations in SIROCCO supported retrieval and explanation but required a constrained transcription language and manual encoding that limited the size of the case base. This section presents a formal case representation and an ontology-constrained extraction pipeline that populates it from unstructured case narratives at scale.

3.1 The Nine-Component D-Tuple

We define a professional ethics case as a D-tuple:

$$D = (R, P, O, S, Rs, A, E, Ca, Cs) \tag{1}$$

The framework synthesizes nine concepts identified across the computational ethics literature into a unified case representation [26]. Each component has a formal definition in the ProEthica core ontology, which constrains what the LLM extracts from case narratives. Table 1 presents the specification for each component, grouped by extraction step. Steps build on entities identified in prior steps. Normative extraction references the roles and states already established, and temporal extraction links actions to the obligations and capabilities already identified.

The extraction pipeline classifies each entity as exactly one component type. Some component pairs are inherently disjoint through their mappings to the Basic Formal Ontology (BFO) [3]. For components that share a BFO parent class, the distinction is enforced through ontology definitions and can be validated through optional human review (Section 3.4). Each extraction step queries the ontology server (Section 3.3) to retrieve existing class definitions before the LLM processes the case narrative, and the LLM provides an ontology match decision

Table 1. Nine-component D-tuple specifications. Components are grouped by extraction step (shaded). Definitions are drawn from the ProEthica core ontology. Cross-component linkages shown in parentheses.

Component	Definition	Key Extraction Fields
Step 1: Contextual Grounding		
Roles (R)	Positions generating ethical obligations	`role_category` [18]; `obligations_generated` ($R \rightarrow O$); `professional_scope`
States (S)	Conditions affecting ethical decisions	`persistence_type` [10]; `obligation_activation` ($S \rightarrow O$); `principle_transformation` ($S \rightarrow P \rightarrow O$)
Resources (Rs)	Knowledge artifacts informing analysis	`resource_category` [21,14]; `authority_source`; `extensional_function`
Step 2: Normative Structure		
Principles (P)	Ethical guidelines establishing professional ideals	`extensional_examples` [21]; `derived_obligations` ($P \rightarrow O$); `potential_conflicts`
Obligations (O)	Required actions or restraint from professional codes	`obligation_type`; `derived_from_principle` ($P \rightarrow O$); `nspe_reference` [13]
Capabilities (Ca)	Competencies enabling actions and fulfilling obligations	`capability_category` [29]; `enables_actions` ($Ca \rightarrow A$); `required_for_obligations`
Constraints (Cs)	Limitations on permissible professional actions	`constraint_type`; `flexibility` [15]; `violation_impact`
Step 3: Temporal Dynamics		
Actions (A)	Volitional interventions by an identified agent	`volitional_nature` [27]; `obligations_fulfilled` ($A \rightarrow O$); `temporal_constraints`
Events (E)	Occurrences outside agent control affecting evaluation	`automatic_nature` [10]; `constraint_activation` ($E \rightarrow Cs$); `state_transitions` ($E \rightarrow S$)

for each entity indicating whether it matches an existing class or represents a novel discovery.

Extraction operates at two levels. Classes represent reusable concept types that apply across cases (e.g., "Disclosure Obligation" as a category). Individuals represent case-specific instances of those classes (e.g., Engineer A's obligation to disclose structural concerns to the municipal review board during the permit phase). New classes proposed by the LLM are flagged for optional review before entering the ontology. Individuals are grounded in specific case text with identified actors and temporal references.

3.2 Three-Step Extraction Methodology

The extraction pipeline transforms unstructured case narratives into the nine-component D-tuple through three sequential steps (Figure 1). Each step builds on previously established context, and all steps query the ontology server for existing class definitions before extraction begins.

Step 1 (Contextual Grounding) extracts Roles, States, and Resources. The pipeline processes Roles first because both States and Resources reference the professional positions that Roles establish, then extracts States and Resources concurrently.

Step 2 (Normative Requirements) extracts Principles, Obligations, Capabilities, and Constraints. The pipeline processes Principles first, then Obligations (which derive from Principles per Dennis et al.'s [13] $P \rightarrow O$ transformation), then Capabilities and Constraints concurrently. This ordering reflects Oakley and Cocking's [24] analysis of role-generated obligations.

Step 3 (Temporal Dynamics) extracts Actions and Events with temporal relations formalized through Allen's [1] interval algebra, mapped to OWL-Time [12]. Actions are evaluated against the Obligations identified in Step 2, and Events are linked to the States and Constraints they activate or transform.

3.3 Ontology Integration via Model Context Protocol

Unconstrained LLM extraction from professional ethics narratives risks producing inconsistent ethical concepts. Schema-guided extraction constrains output format but not conceptual vocabulary. ProEthica addresses both levels by injecting formal concept definitions from a domain ontology into the LLM context at extraction time.

OntServe,[1] a dedicated ontology server, exposes the ProEthica domain ontology through a SPARQL endpoint accessible via the Model Context Protocol (MCP). MCP makes the ontology available to the LLM as a set of callable tools. The pipeline retrieves existing class definitions for the relevant component types and injects them into the prompt before each extraction step. The LLM can then issue additional MCP tool calls during extraction to query class hierarchies, check

[1] https://github.com/cr625/OntServe

whether a proposed concept already exists in the ontology, or retrieve related definitions.

For example, when extracting Obligations, the pipeline injects existing obligation class definitions along with Roles and Principles from earlier steps. The LLM queries the ontology to check for matching classes or propose novel concepts. OntServe checks proposed concepts for near-duplicates and responds with an exact match, a set of candidates, or acceptance as a new class. The constraint is cumulative. Each committed case expands the vocabulary available to all subsequent extractions.

3.4 Case Synthesis and Review

After the three extraction steps produce the nine-component representation, a synthesis phase analyzes the Questions, Conclusions, and References sections of each NSPE BER case using the extracted entities as structured context. Synthesis produces six outputs. Code provision mapping (linking obligations and principles to NSPE Code sections), question and conclusion identification (capturing the board's framing and resolution), precedent case extraction, transformation classification (categorizing how the case altered existing standards), causal-normative analysis (tracing how events triggered obligation activation and produced resolution patterns), and decision point synthesis.

The pipeline supports two execution modes. In fully automated mode, all proposed entities and ontology classes are accepted at their default values, and extraction through synthesis completes without intervention. A step-by-step mode exposes an entity review interface where domain experts can accept or reject proposed classes before they are committed, a workflow designed primarily for initial ontology construction when practitioners integrate existing ontologies with imported ethical guidelines.

Accepted classes are written to the OntServe ontology and become available as vocabulary for all subsequent extractions, so the case base and ontology co-evolve as cases are processed.

3.5 Implementation

All three extraction steps use a parameterized extractor that calls the Anthropic API with concept-specific prompt templates stored in the database. A separate LangGraph state graph supports extended temporal dynamics analysis through coordinated stages including temporal marker identification, action and event extraction, causal chain analysis, and timeline construction. ProEthica employs a dual-database architecture. OntServe maintains the domain ontology in OWL/Turtle files organized into three layers (core component classes, domain-specific subclasses, per-case individuals) and exposes its contents through MCP. A PostgreSQL database augmented with pgvector stores document sections and per-component feature vectors as 384-dimensional embeddings generated by

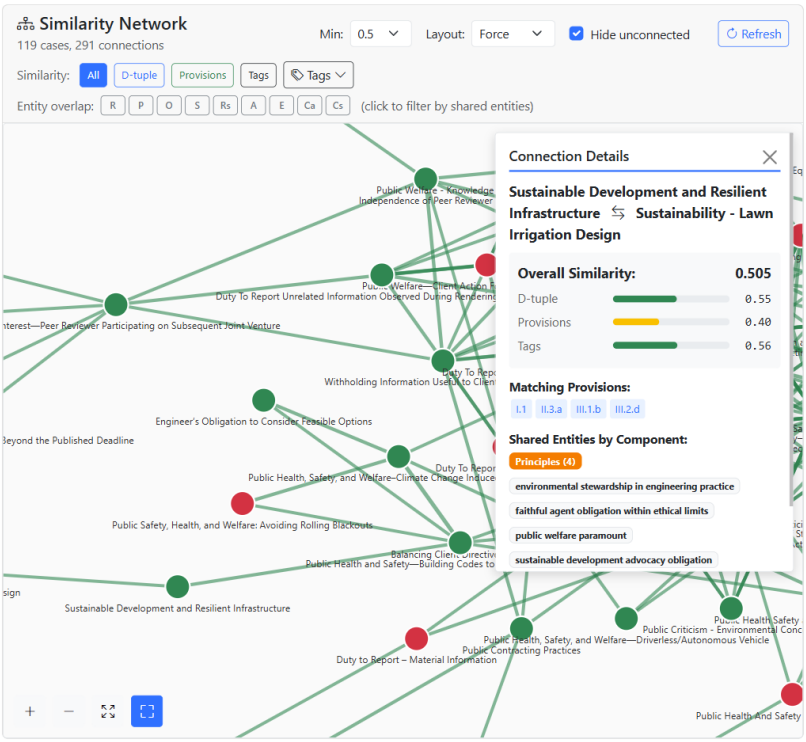

Fig. 2. Similarity network (119 cases, minimum similarity 0.5). Nodes colored by outcome; detail panel shows provisions, entity count, and connected cases with scores. Entity overlap filters (R, P, O, S, Rs, A, E, Ca, Cs) let users identify which components two cases share and filter connections by component type.

all-MiniLM-L6-v2 [30]. For each component type in a case, the system concatenates entity labels and definitions and embeds the result. These per-component embeddings support the similarity computation described in Section 4.

The current case base contains 119 NSPE Board of Ethical Review cases spanning 1958 to 2025, each with section embeddings, provision extraction, outcome classification, subject tags, and nine-component D-tuple embeddings generated through the full extraction pipeline described above. A similarity network interface (Figure 2) visualizes case relationships and provides component-level entity overlap filters (one per D-tuple component), allowing users to examine which extracted entities two cases share and to trace specific correspondences through the ontology.

4 Component-Aware Case Retrieval

Embedding-based retrieval offers scalability for case comparison, but its effectiveness for professional ethics cases has not been validated against expert-authored citation patterns. This section evaluates whether the multi-factor similarity

approach in ProEthica recovers cases that NSPE board members cite as precedent in their published opinions, whether replacing document-level embeddings with component-aware embeddings improves retrieval by preserving the D-tuple structure, and whether the resulting component-level similarity profiles capture domain-relevant variation across case topics.

BER opinions contain a Facts section presenting the scenario and a Discussion section containing the board's ethical analysis. The baseline computation embeds these sections separately, producing two vectors per case. The D-tuple methods described below replace these with embeddings derived from extracted component entities.

4.1 Multi-Factor Similarity Architecture

ProEthica computes case similarity as a weighted combination of text-based embedding and set-based feature comparisons. All three embedding strategies share the same multi-factor scoring formula, with 40% of the overall score allocated to text-based embedding and 60% to four set-based features, including provision overlap ($w = 0.25$, Jaccard over cited NSPE Code sections), outcome alignment ($w = 0.15$, ternary: ethical, unethical, mixed), tag overlap ($w = 0.10$), and principle ethical tension overlap ($w = 0.10$). Only the computation within the 0.40 embedding allocation differs across methods, so any retrieval difference is attributable solely to the embedding strategy.

4.2 Three Embedding Strategies

Section-based embedding (baseline). The baseline embeds the Facts and Discussion sections of each case separately using all-MiniLM-L6-v2 [30], producing two 384-dimensional vectors per case. Cosine similarity between corresponding section embeddings yields two scores, weighted at 0.15 (facts) and 0.25 (discussion), for a combined textual contribution of 0.40.

Combined D-tuple embedding. The combined method takes the same nine component embeddings described below and combines them via weighted average (using the weights in Table 2) into a single L2-normalized vector per case. At retrieval time, a single cosine similarity between these vectors occupies the 0.40 embedding allocation. This method captures the D-tuple formalization (entities are extracted and typed by component) but collapses component-level distinctions into a single embedding space before comparison.

Per-component embedding. The per-component method preserves component independence through retrieval. For each of the nine D-tuple component types in a case, the system concatenates all entity labels and definitions of that type and generates a separate L2-normalized embedding. At retrieval time, nine independent cosine similarities are combined by weighted sum:

$$sim_{comp}(D_i, D_j) = \frac{\sum_{k=1}^{9} w_k \cdot \cos(\mathbf{e}_{i,k}, \mathbf{e}_{j,k})}{\sum_{k=1}^{9} w_k} \tag{2}$$

where w_k is the weight assigned to component k and $\mathbf{e}_{i,k}$ is the embedding vector for component k of case i. Each component occupies an independent semantic channel, so that high similarity in Principles paired with low similarity in Events is preserved rather than averaged away. The resulting per-component score occupies the same 0.40 embedding allocation as the combined method.

The three methods differ in how much D-tuple structure they preserve. Section-based embedding uses no D-tuple information. Combined D-tuple embedding uses the extraction but merges components before comparison. Per-component embedding preserves component independence through the full retrieval pipeline.

The weights in Table 2 are initial estimates based on the assumed salience of each component type in professional ethics evaluation. As Section 4.5 will show, the assumption that boundary and competency components (Constraints and Capabilities) are least discriminative does not hold uniformly. Learning weights from expert similarity judgments is planned future work.

Table 2. Component weights for per-component similarity. Weights are baseline heuristics; calibration against expert judgments is future work.

Component	Weight	Component	Weight
Principles (P)	0.20	Resources (Rs)	0.10
Obligations (O)	0.15	Actions (A)	0.10
Roles (R)	0.12	Constraints (Cs)	0.08
States (S)	0.10	Events (E)	0.08
		Capabilities (Ca)	0.07

When a case lacks entities for a component type, the remaining weights are renormalized to sum to 1.0. Component text is truncated at 2,000 characters before embedding. All three methods use the same embedding model to isolate the effect of embedding granularity from model choice.

4.3 Citation Ground Truth Validation

NSPE BER opinions routinely cite prior board decisions as precedent. These expert-authored citations provide ground truth for evaluating retrieval quality. We extracted citation edges from the 119-case pool, yielding 228 directed citation links across 93 source cases (mean 2.5 citations per case). The remaining 26 cases predate the board's practice of citing prior opinions and contain no outgoing citation edges. For each source case, we ranked the remaining 118 candidates by the multi-factor similarity score and measured the rank positions of the expert-cited cases. Mean Reciprocal Rank (MRR) measures how early the first cited case appears. Recall@K measures the fraction of cited cases recovered within the top K results.

All three methods recover cited precedents at 3 to 8 times the random baseline, depending on the metric (Table 3). Both D-tuple methods outperform the section baseline across all four metrics. The two D-tuple aggregation strategies perform comparably, with combined achieving the highest R@10 (0.463) and per-component the highest R@5 (0.331). These metrics are conservative lower

Table 3. Citation ground truth retrieval across 93 source cases with 228 citation edges (118 candidates per query). All methods share the same multi-factor formula (0.40 embedding + 0.60 set-based features). Only the embedding computation differs.

Metric	Section	Combined	Per-comp.	Random
MRR	0.380	**0.407**	0.403	0.086
Recall@5	0.286	0.324	**0.331**	0.042
Recall@10	0.420	**0.463**	0.442	0.085
Recall@20	0.536	**0.577**	0.554	0.169

bounds because board members cite the most salient precedents, not all related cases.

Citation text ablation. BER Discussion sections describe cited precedents, which could inflate section-based similarity. An ablation that strips citation-containing sentences (326 of 3,932 sentences) and re-embeds yields no systematic inflation. Scored with section embedding alone (set-based features zeroed), the stripped condition achieves R@10 of 0.397 versus 0.387 for the original, with mixed per-edge effects (107 edges improved, 80 worsened, 41 unaffected). D-tuple methods are structurally immune to this confound because they embed extracted entity content, not raw case text.

Embedding isolation ablation. Because all three methods share 60% of their score from identical set-based features, the observable embedding effect is compressed. An ablation that scores cases using only the embedding component (set-based features zeroed) reveals a larger D-tuple advantage. Per-component embedding alone achieves Recall@10 of 0.568, compared to 0.387 for section-based embedding alone ($\Delta = +0.181$). Set-based features alone achieve 0.387. Per-component embedding alone (0.568) outperforms its own full-formula result (0.442), indicating that the 0.60 set-feature weight dilutes the D-tuple embedding signal at this configuration.

4.4 Rank Correlation Analysis

To assess how the three methods relate beyond citation retrieval, we computed pairwise Spearman rank correlation and top-10 overlap for all 119 cases across the full 118-candidate rankings (Table 4).

Table 4. Pairwise agreement across 119 NSPE BER cases. Spearman ρ is averaged across per-case rank correlations over the full 118-candidate ranking. Overlap@10 measures the fraction of shared cases in the top-10 results.

Metric	Sec. vs Comb.	Sec. vs Comp.	Comb. vs Comp.
Mean Spearman ρ	0.929	0.935	0.991
Mean Overlap@10	76.9%	78.3%	91.5%

The two D-tuple methods produce nearly identical rankings ($\rho = 0.991$, Overlap@10 = 91.5%), confirming that the D-tuple extraction formalization is the primary contributor to retrieval quality, not the aggregation strategy.

Both D-tuple methods diverge from section-based ranking by a similar degree ($\rho \approx 0.93$, Overlap@10 $\approx$ 77–78%), indicating that the transition from section text to D-tuple content produces a consistent shift in how cases are ranked.

4.5 Analysis of Divergent Cases

The three cases with the lowest section-vs-per-component ρ are Case 19 (Duty to Report Misconduct, $\rho = 0.711$), Case 105 (Reviewing Work of Another Engineer, $\rho = 0.785$), and Case 141 (Selection of Firm, $\rho = 0.821$). Per-component similarity profiles for the top-10 neighbors of each case reveal which components drive the ranking disagreement.

Case 19 illustrates the pattern. Actions shows the highest variance among top-10 neighbors (std = 0.105, range 0.30–0.72), while Constraints shows the lowest (std = 0.038). A case about reporting misconduct shares general ethical constraints with many cases, but the professional actions taken vary widely among neighbors. Across all three divergent cases, Actions is consistently among the highest-variance components (std = 0.073–0.105) and Capabilities among the lowest (std = 0.025–0.060), but the most discriminative component varies. Roles dominates in Case 105, Principles in Case 141. Section-based embedding buries these component-level distinctions in two monolithic vectors. The per-component method preserves them, and the ranking divergences occur in cases where the discrimination pattern is most case-specific.

4.6 Connection to CBR Retrieval Literature

The component-aware approach occupies a middle position between flat document retrieval and fully symbolic case matching. The embedding isolation ablation provides empirical evidence for this positioning. Per-component embedding alone recovers R@10 of 0.568, compared to 0.387 for section-based embedding alone, confirming Wiratunga et al.'s [32] finding that weighted retrieval across structured case components outperforms flat document retrieval. The divergent-case analysis extends this result by showing that the most discriminative component varies by case, a pattern consistent with Sun et al.'s [28] demonstration that structured legal features improve retrieval over surface-level matching. ProEthica adds ontology grounding to both ideas, constraining the vocabulary from which component entities are drawn.

5 Discussion

5.1 What the Experiments Reveal

Because the weight configuration and set-based features are identical across methods, any retrieval difference is attributable to the embedding strategy alone. Both D-tuple methods outperform section-based embedding across all metrics (Table 3), and the near-perfect rank correlation between them ($\rho = 0.991$, Table 4)

confirms that the D-tuple extraction formalization is the primary contributor, not the aggregation strategy. At R@10, examining fewer than 9% of the candidate pool recovers nearly half of all expert-cited precedents, reducing the manual search space by an order of magnitude. The embedding isolation ablation reveals that the 0.40 embedding weight underestimates the D-tuple contribution. D-tuple embeddings are strong enough to be diluted by set features, while section-based embeddings are weak enough to benefit from them.

The divergent-case analysis (Section 4.5) shows that the most discriminative component varies by case. Actions dominates in Case 19, Roles in Case 105, Principles in Case 141. Section-based embedding merges these distinct discrimination patterns into two monolithic vectors, while the per-component method preserves them as separate channels.

These retrieval results reflect a broader design property of structured case representation. A section-based similarity score offers no explanation for why two cases are similar. The per-component method produces a nine-dimensional similarity profile that identifies which components two cases share and where they differ, mirroring how ethics boards compare cases on specific features rather than overall impression. Component-level representation also enables targeted retrieval queries, such as cases with similar Obligations but different temporal profiles, that a monolithic embedding cannot express. Presenting component profiles for examination rather than ethical verdicts follows the hypothesis-driven paradigm that Benk and Miller [8] show preserves stable evidence standards in AI-assisted evaluation.

5.2 Limitations

The 228 citation edges across 93 source cases provide a meaningful evaluation signal, but the pool remains small relative to the full NSPE archive (over 650 cases). The component weights are domain-informed heuristics. The divergent-case analysis suggests that discriminative power varies across components in ways the current weights do not reflect. The embedding model (all-MiniLM-L6-v2) is a general-purpose sentence transformer, not fine-tuned for ethics or legal text. The traceability of similarity scores to extracted entities has not been evaluated with domain experts. Optimizing the weight configuration and the embedding model, grounded in expert similarity judgments, is future work.

6 Conclusion and Future Work

ProEthica demonstrates that ontology-constrained LLM extraction can populate nine-component structured case representations from professional ethics narratives, with similarity scores traceable to the specific extracted entities that produced them (Figure 2). The near-perfect rank correlation between the two D-tuple strategies ($\rho = 0.991$) identifies the extraction formalization as the primary contributor. Future work includes weight calibration against expert similarity judgments, evaluation of component-level profiles with domain experts, extension to education and legal ethics domains, and scenario analysis for user-supplied vignettes where board conclusions are not yet available.

Acknowledgments. The NSPE Board of Ethical Review cases used in this study are publicly available through the NSPE Ethics Resources website.

Declaration on Generative AI During the preparation of this work, the authors used Claude (Anthropic) to assist with manuscript editing and experiment analysis scripts. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

Disclosure of Interests. The authors have no competing interests to declare.

Data and Code Availability Experiment data, analysis scripts, and the ProEthica source code are available at https://github.com/cr625/proethica (tag: `iccbr-2026-experiments`).

References

1. Allen, J.F.: Maintaining knowledge about temporal intervals. Communications of the ACM **26**(11), 832–843 (1983)
2. American Law Institute: Restatement of the Law Third, Torts: Liability for Physical and Emotional Harm. American Law Institute Publishers (2010)
3. Arp, R., Smith, B., Spear, A.D.: Building Ontologies with Basic Formal Ontology. MIT Press (2015)
4. Ashley, K.D.: Reasoning with cases and hypotheticals in HYPO. International Journal of Man-Machine Studies **34**(6), 753–796 (1991)
5. Ashley, K.D., McLaren, B.M.: A CBR knowledge representation for practical ethics. In: Advances in Case-Based Reasoning. pp. 180–197. Springer (1995)
6. Bach, K., Bergmann, R., Brand, F., Caro-Martínez, M., Eisenstadt, V., et al.: Case-based reasoning meets large language models: A research manifesto for open challenges and research directions. https://hal.science/hal-05006761 (2025)
7. Bench-Capon, T.J.M., Sartor, G.: A model of legal reasoning with cases incorporating theories and values. Artificial Intelligence **150**(1-2), 97–143 (2003)
8. Benk, M., Miller, T.: Same performance, hidden bias: Evaluating hypothesis- and recommendation-driven AI. arXiv preprint arXiv:2603.15824 (2026)
9. Bergmann, R., Brand, F., Lenz, M., Malburg, L.: EXAR: A unified experience-grounded agentic reasoning architecture. In: Case-Based Reasoning Research and Development. pp. 3–17. Springer (2025)
10. Berreby, F., Bourgne, G., Ganascia, J.G.: A declarative modular framework for representing and applying ethical principles. In: 16th Conference on Autonomous Agents and MultiAgent Systems. pp. 96–104 (2017)
11. Brand, F., Malburg, L., Bergmann, R.: Large language models as knowledge engineers. In: ICCBR 2024 Workshop Proceedings (2024)
12. Cox, S., Little, C.: OWL-Time: Time ontology in OWL. W3c recommendation, W3C (2017)
13. Dennis, L.A., Fisher, M., Slavkovik, M., Webster, M.: Formal verification of ethical choices in autonomous systems. Robotics and Autonomous Systems **77**, 1–14 (2016)
14. Frankel, M.S.: Professional codes: Why, how, and with what impact? Journal of Business Ethics **8**(2-3), 109–115 (1989)
15. Ganascia, J.G.: Ethical system formalization using non-monotonic logics. In: Proceedings of the Annual Meeting of the Cognitive Science Society. vol. 29 (2007)

16. Ge, X., Xu, J.: Integrating case-based reasoning with LLM for expense fraud detection. In: Case-Based Reasoning Research and Development. pp. 52–66. Springer (2025)
17. Ghazouani, F., Giustozzi, F., Le Ber, F.: LLM-driven case-base populating for structuring and integrating restoration experiences. In: Case-Based Reasoning Research and Development. pp. 67–80. Springer (2025)
18. Kong, K.C.C., Lam, P.W.Y., Cheng, W.: Corpus-based empirical approach to professionalism: Identifying interactional roles and dispositions in professional codes of ethics. Journal of Applied Linguistics and Professional Practice **14**(1), 3–28 (2020)
19. Lenz, M., Hoffmann, M., Bergmann, R.: LLsiM: Large language models for similarity assessment in case-based reasoning. In: Case-Based Reasoning Research and Development. pp. 126–141. Springer (2025)
20. Levi, E.H.: An Introduction to Legal Reasoning. University of Chicago Press (1949)
21. McLaren, B.M.: Extensionally defining principles and cases in ethics: An ai model. Artificial Intelligence **150**(1-2), 145–181 (2003)
22. Mumford, J., Atkinson, K., Bench-Capon, T.: Unravelling the ECHR: Components of legal case analysis. In: JURIX 2024. vol. 395, pp. 107–118 (2024)
23. National Society of Professional Engineers: Board of ethical review case archive. https://www.nspe.org/career-growth/ethics/board-ethical-review-cases (2025), accessed March 2026
24. Oakley, J., Cocking, D.: Virtue Ethics and Professional Roles. Cambridge University Press (2001)
25. Rauch, C.B.: Precedent-based professional role ethics for AI decision analysis. In: Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society (AIES). vol. 8, pp. 2921–2923 (2025)
26. Rauch, C.B., Weber, R.O.: ProEthica: A professional role based ethical analysis tool using LLM-orchestrated, ontology supported case based reasoning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 40, pp. 41673–41675 (2026). https://doi.org/10.1609/aaai.v40i48.42377
27. Sarmiento, C., Bourgne, G., Inoue, K., Ganascia, J.G.: Action languages based actual causality in decision making contexts. In: PRIMA 2022: Principles and Practice of Multi-Agent Systems. pp. 243–259. Springer (2023)
28. Sun, Z., Zhang, K., Yu, W., Wang, H., Xu, J.: Logic rules as explanations for legal case retrieval. In: LREC-COLING 2024. pp. 10747–10759. ELRA and ICCL (2024)
29. Tolmeijer, S., Kneer, M., Sarasua, C., Christen, M., Bernstein, A.: Implementations in machine ethics: A survey. ACM Computing Surveys **53**(6), 132:1–132:38 (2020)
30. Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., Zhou, M.: MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In: Advances in Neural Information Processing Systems. vol. 33, pp. 5776–5788 (2020)
31. Wilkerson, K., Leake, D.: Case hallucinations and steps toward repair. In: CBR-LLM Workshop at ICCBR 2025. pp. 43–54 (2025)
32. Wiratunga, N., et al.: CBR-RAG: Case-based reasoning for retrieval augmented generation in LLMs for legal question answering. In: Case-Based Reasoning Research and Development. pp. 445–460. Springer (2024)
33. van Woerkom, W., Grossi, D., Prakken, H., Verheij, B.: A fortiori case-based reasoning: From theory to data. Journal of Artificial Intelligence Research **81**, 401–441 (2024)
34. Zhong, T., Song, Y., Limarga, R., Pagnucco, M.: Computational machine ethics: A survey. Journal of Artificial Intelligence Research **82**, 1581–1628 (2025)