RPI

Funding Acknowledgement

A Benchmark for Gap and Overlap Analysis as a Test of KG Task Readiness

Maruf Ahmed Mridul[1]
mridum@rpi.edu

Rohit Kapa[2]
rohit.kapa@prudential.com

Oshani Seneviratne[1]
senevo@rpi.edu

[1]Rensselaer Polytechnic Institute
[2]Prudential Financial, Inc.

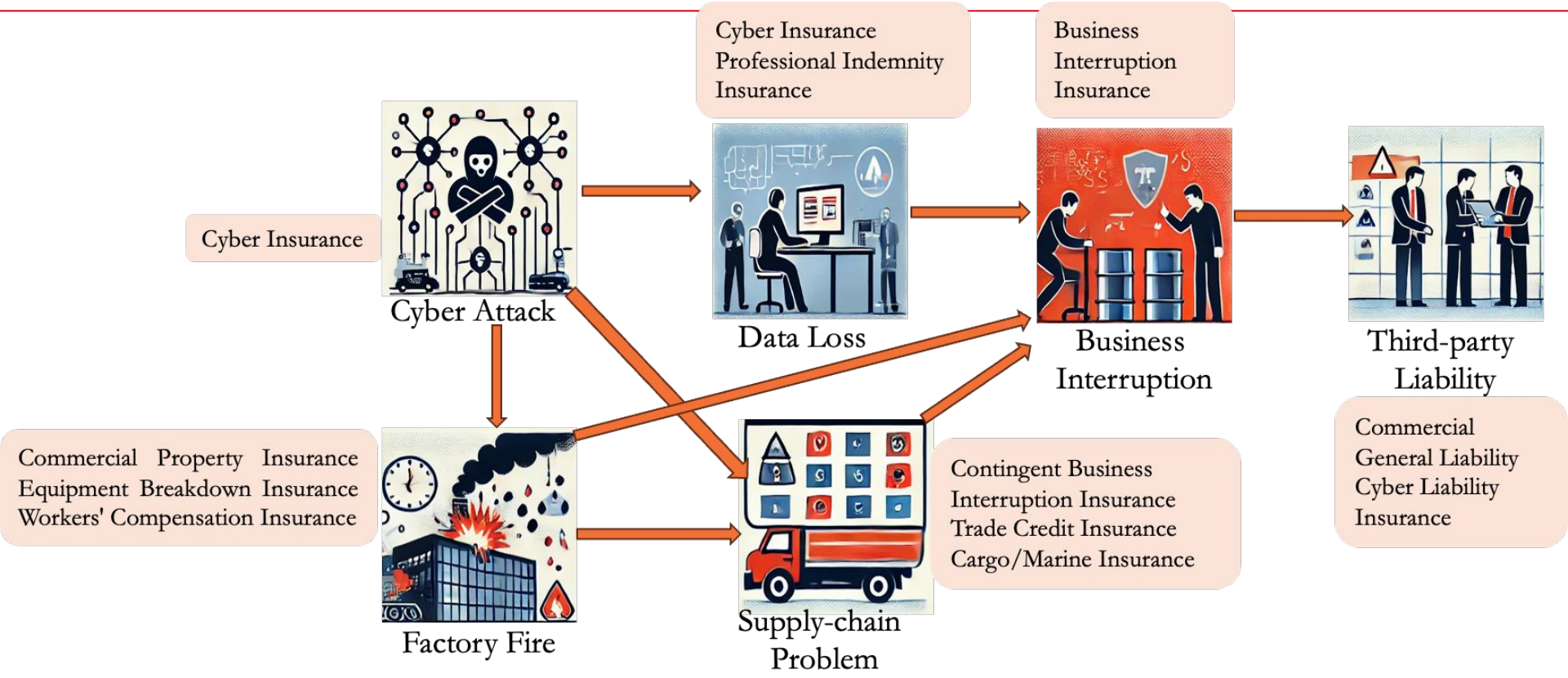
QKG@ESWC 2026

May 11, 2026

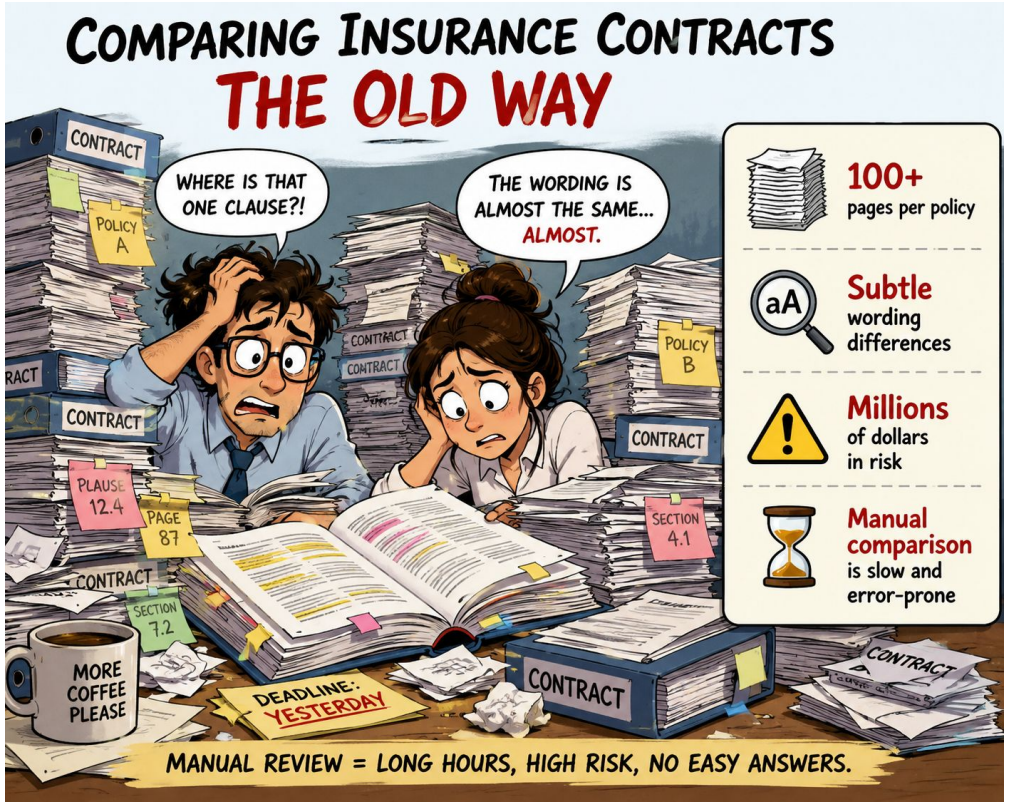
Core Research Question

Can a Knowledge Graph actually perform the reasoning task it was designed for?

Complex business insurance scenario involving multiple policies



Use Case



Gap and Overlap Analysis

For a given real-world scenario, which contracts:

- **Cover** the event?
- **Deny** the event?
- **Do Not Apply**?

Definitions

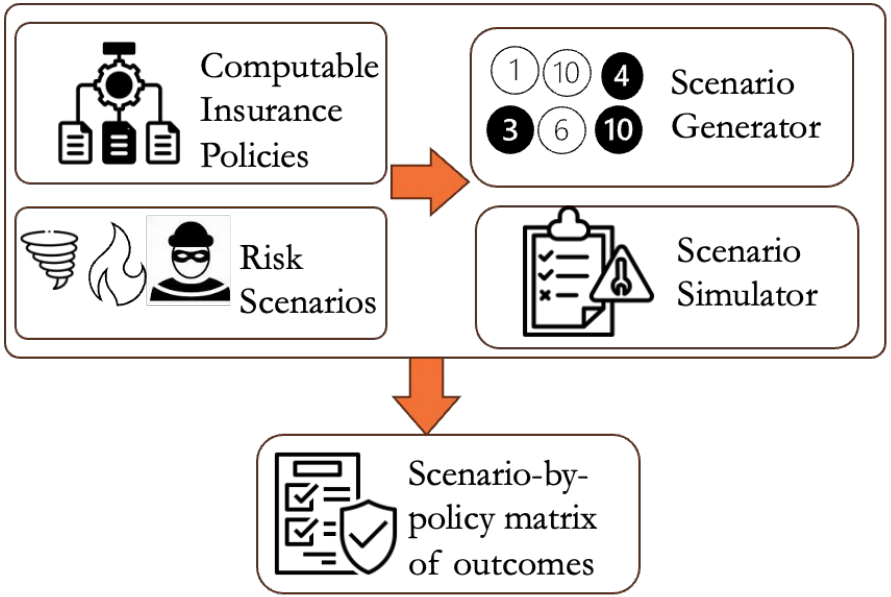
- **Overlap**: Multiple contracts provide coverage for the same scenario.
- **Gap**: No contract provides coverage.
- **Conflict**: Contracts differ in how they treat the same event.

Why do we need this?

- Reveals missing protection
- Identifies redundant coverage
- Supports product comparison and harmonization
- Requires precise, rule-based reasoning

An Insurance KG is Task-Ready if it can:

- Interpret a scenario
- Determine coverage or denial
- Compare outcomes across contracts
- Provide clause-level evidence
- Produce consistent, deterministic answers



Contributions

- A manually curated corpus of **ten simplified but diverse** life insurance contracts, verified by a domain-expert.
- A domain **ontology (TBox)** that formalizes the main contract concepts and relations, together with an **aligned knowledge graph (ABox)** populated from the contract facts, enabling deterministic, query-based analysis.
- A suite of **58 structured scenarios** and a **corresponding ground truth** for **gap and overlap analysis**.
    - For each scenario, we include **contract-level outcomes** and c**lause-level source excerpts** that justify those outcomes, supporting traceability and diagnosis.
- Evaluation:
    - **A text-only LLM pipeline** that infers contract responses for each scenario,
    - **An ontology-driven pipeline** that answers the same scenarios using SPARQL over the instantiated ontology.

The Contracts

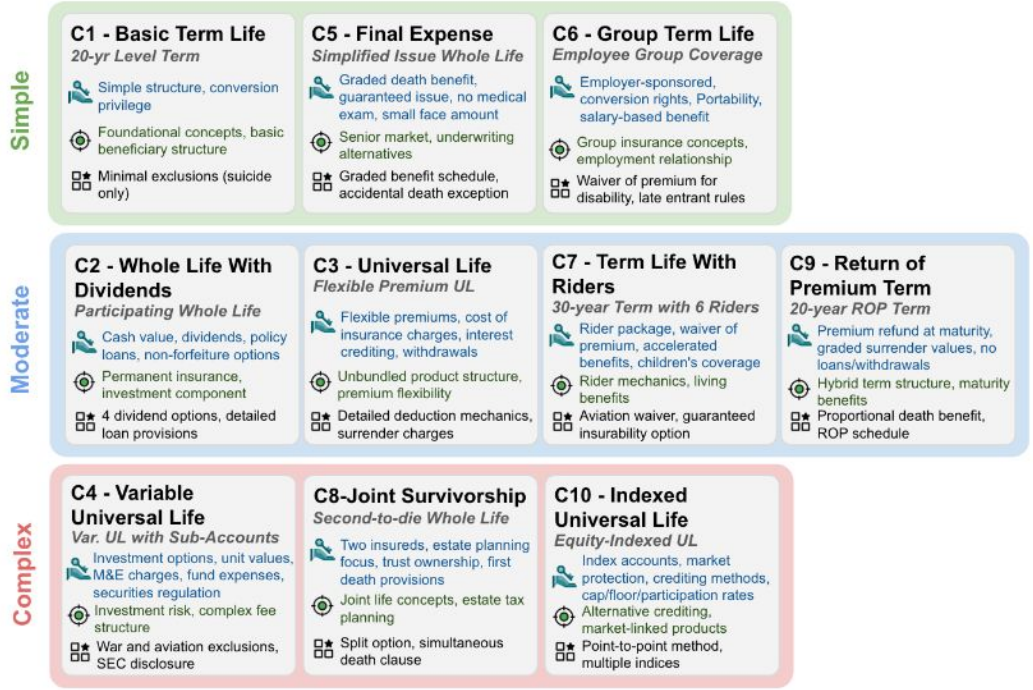


Figure 1: Overview of the 10 life insurance contracts categorized by complexity. The contracts are grouped into three levels: *Simple*, *Moderate*, and *Complex*, based on their complexity and range of features. , , and represent the *Key Features*, *Focus*, and *Uniqueness*, respectively. C1-C10 are the identifiers of the contracts.

This progression is important because it allows us to test whether both the knowledge graph and LLMs can handle increasingly sophisticated contractual structures.

The Ontology

```
li:SuicideExclusion a owl:Class ; rdfs:subClassOf li:Exclusion ;
    rdfs:label "Suicide Exclusion" ;
    rdfs:comment "Present in ALL 10 contracts. Period: 12 months (C6 only); 24 months (
        C1,C2,C3,C4,C5,C7,C8,C9,C10). Benefit outcome varies - use li:suicideBenefitType
         object property with li:SuicideBenefitType individuals." .
```

```
li_abox:SuicideExclusion_C1 a li:SuicideExclusion ;
    li:suicideExclusionPeriodMonths 24 ;
    li:suicideBenefitType li:ReturnPremiumsPaid ;
    li:sourceContractID "C1" ;
    li:sourceContractSection "Section 7.1" ;
    li:coverageSourceText "SUICIDE CLAUSE: If the insured commits suicide within two
        years from the issue date, our liability is limited to a refund of premiums paid
        ." ; .
```

The Ontology - Qualities

- Class hierarchy is organized around the **subsumption pattern**
- Where a concept admits a fixed set of mutually exclusive values, the TBox applies the **value partition pattern**
- Maintains a clean **TBox/ABox separation**
- Object and datatype properties carry explicit **rdfs:domain** and **rdfs:range** declarations
- Dedicated datatype properties defined in the TBox: **sourceContractID, sourceContractSection, and coverageSourceText**.
    - Helps with traceability

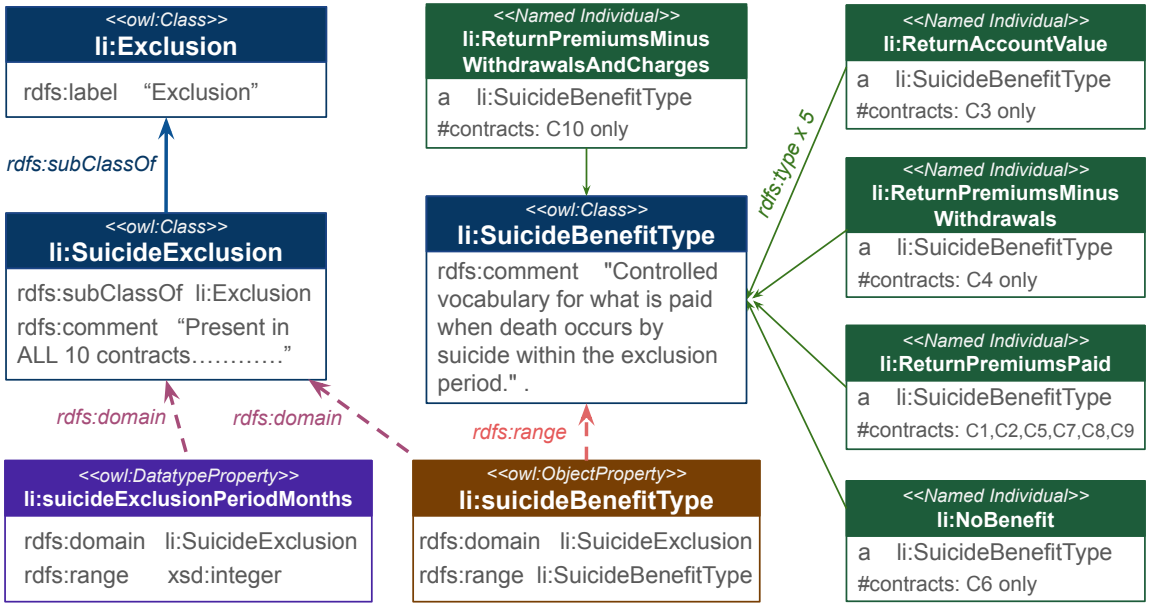


Figure 2: A representative TBox excerpt

The Ontology - Coverage

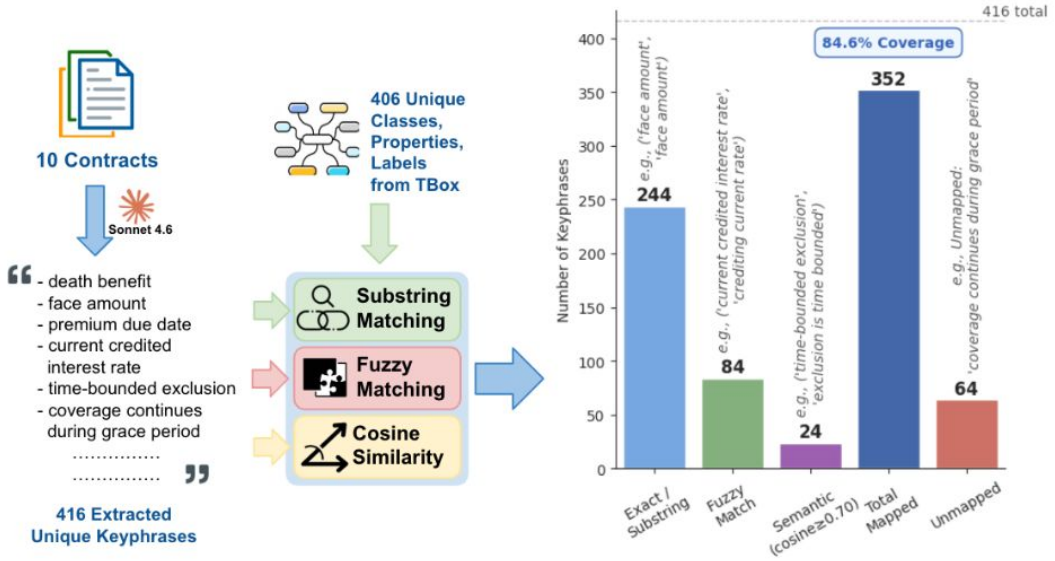


Figure 3: Ontology Coverage Analysis. Extracted keyphrases from the contracts are matched against the ontology TBox aritfacts (classes, properties, labels) using three different matching techniques. Example tuples (s, o) denote the **source contract term** (s) and its corresponding **mapped ontology artifact** (o).

Manual inspection shows that the **unmatched keyphrases** do **not** necessarily represent **gaps** in the ontology, rather they are **missed** by the **matching method**

li:PremiumNonpaymentGracePeriod a **owl**:Class;
rdfs:comment *"Triggered by failure to pay a scheduled premium. Coverage continues for specified days after premium due date. C1: 31 days, C2: 31 days, C5: 30 days, C7: 31 days . . . , C9: 31 days."*

Scenario Suite

- **Scenario Description:** A natural language prompt detailing a specific insurance event (e.g., *SCEN-003: "Insured dies by suicide exactly 13 months after the policy issue date. All premiums paid on time."*).
- **Double-Query Ground Truth:** Two distinct SPARQL queries, one to identify contracts where the claim is covered and one where it is denied.
- **Contract-Level Outcomes:** Manually verified ground truth for each of the ten contracts (C1–C10), detailing the specific status: COVERED, DENIED, or NOT_APPLICABLE.
- **Traceability Evidence:** For every outcome, we include the verbatim *sourceText* and *sourceSection* from the original contract to ensure results are auditable and explainable.


```
{
  "scenario_id": "SCEN-008",
  "category": "POLICY_LAPSE_NON_FORFEITURE",
  "scenario_description": "After several years, the policy owner stops paying premiums and neither surrenders the policy nor selects
  "query_covered": "PREFIX li: <http://www.lifeinsurance.org/ontology#> SELECT DISTINCT ?contractID ?optionType ?sourceSection ?sour
  "query_denied": "PREFIX li: <http://www.lifeinsurance.org/ontology#> SELECT DISTINCT ?contractID ?sourceSection ?sourceText WHERE
  "contract_responses": [
    {
      "contract_id": "C1",
      "status": "NOT_APPLICABLE",
      "details": "Term life policy with no cash value. Upon lapse, coverage simply terminates. No non-forfeiture options exist.",
      "source": "Section 6.3: 'If the premium is not paid by the end of the grace period, the policy will lapse and coverage will te
    },
    {
      "contract_id": "C2",
      "status": "COVERED",
      "details": "After 3+ years, Extended Term Insurance automatically applies if no election is made within 60 days of lapse. The
      "source": "Section 9.3, 9.4: 'EXTENDED TERM INSURANCE: Continue the full face amount for a limited period based on cash value.
    },
    {
      "contract_id": "C3",
      "status": "COVERED",
      "details": "Upon lapse, any Account Value in excess of surrender charges is automatically paid to the policy owner. No traditi
      "source": "Article 10.3: 'Upon lapse, any Account Value in excess of surrender charges will be paid to you automatically.'"
    },
    {
      "contract_id": "C4",
      "status": "NOT_APPLICABLE",
      "details": "No traditional non-forfeiture options explicitly listed. Upon lapse, any surrender value remaining after account v
      "source": "Section 7 addresses lapse conditions but does not specify non-forfeiture options beyond surrendering the account va
    },
    {
      "contract_id": "C5",
      "status": "COVERED",
      "details": "After 2+ years, the policy automatically continues with a reduced benefit amount based on the accumulated cash val
      "source": "Section 8.1: 'After 2 years, if you stop paying premiums, your policy will automatically continue with a reduced be
```

Gap and Overlap Analysis

$$\text{Overlap}_{\text{insured}}(s) = \{\, c \in \mathcal{C} \mid s \text{ is } \textsc{Covered} \text{ under } c \,\}$$

$$\text{Gap}_{\text{insured}}(s) = \{\, c \in \mathcal{C} \mid s \text{ is } \textsc{Denied} \text{ or } \textsc{Not\_Applicable} \text{ under } c \,\}$$

- Q_s^+ : a coverage query that retrieves contracts for which the scenario is COVERED
- Q_s^- : a denial query that retrieves contracts for which the scenario is DENIED
- **M** : Scenario-by-contract outcome matrix

$$\begin{aligned} M[s,c] &= \textsc{Covered} && \text{if } c \in \text{Ans}(Q_s^+), \\ M[s,c] &= \textsc{Denied} && \text{if } c \in \text{Ans}(Q_s^-), \\ M[s,c] &= \textsc{Not\_Applicable} && \text{if } c \notin \text{Ans}(Q_s^+) \ \wedge \ c \notin \text{Ans}(Q_s^-). \end{aligned}$$

Executing the scenario queries over all $s \in \mathcal{S}$ yields a scenario-by-contract outcome matrix:

$$M[s,c] \in \{\textsc{Covered}, \textsc{Denied}, \textsc{Not\_Applicable}\}.$$

From M, insured-centric overlap and gap can be computed directly:

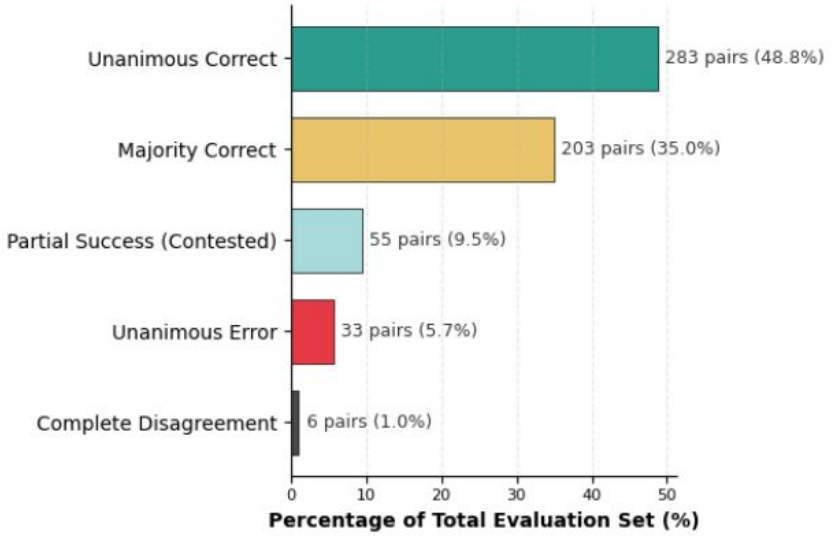
$$\begin{aligned} |\text{Overlap}_{\text{insured}}(s)| &= |\{\, c \in \mathcal{C} \mid M[s,c] = \textsc{Covered} \,\}|, \\ |\text{Gap}_{\text{insured}}(s)| &= |\{\, c \in \mathcal{C} \mid M[s,c] \in \{\textsc{Denied}, \textsc{Not\_Applicable}\} \,\}|. \end{aligned}$$

Results

- The **three most complex** contracts in the benchmark all appear among the **top-five most error-prone** contracts for a **majority** of the evaluated models.
- The quality of **evidence** citations provided by LLMs varies substantially.
- LLMs frequently makes decisions based on it's **prior knowledge** about certain things **rather than** completely relying on **the given data**.

Model	Accuracy
Claude Sonnet 4.6	**87.76%** (509/580)
ChatGPT-5.3	72.93% (423/580)
Gemini-3	65.17% (378/580)
Aggregate	**75.29%** (1310/1740)

Mismatch type (TRUE → PRED.)	Count
NOT_APPLICABLE → DENIED	207
NOT_APPLICABLE → COVERED	126
COVERED → DENIED	30
COVERED → NOT_APPLICABLE	30
DENIED → COVERED	23



Gap/Overlap Analysis as a Test of KG Task Readiness

- Shifts evaluation from "data completeness" to "analytical utility": can the KG handle complex, rule-based comparisons?
- If the ontology-based representation can determine which contracts cover a scenario and which don't, consistently and with clause-level justification, it has proven its task readiness.
- The 58 scenarios function as executable competency questions: passing them means the TBox and ABox are sufficient to support the analytical task they were built for.

Conclusion

- Gap and overlap analysis serves as an effective test of KG task readiness.
- We release an executable, auditable benchmark: 10 expert-verified contracts, a domain ontology (TBox + ABox), and 58 SPARQL-grounded scenarios with clause-level evidence, all fully aligned.
- Code: https://github.com/brains-group/gap_and_overlap_analysis_insurance_contract_benchmark
- Questions?
    - senevo@rpi.edu

Postdoc Opportunity

I am hiring a Postdoctoral Researcher in Agentic AI & Federated Language Models at RPI.

More Info:
https://www.linkedin.com/pulse/postdoctoral-researcher-agentic-ai-federated-language-seneviratne-vppce