PoliCond: Condition-Aware Ontology-Driven LLMs for Privacy Policy Contradiction Analysis

Yalin Feng, Yifei Lu*, Minxue Pan*
State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
Software Institute, Nanjing University, Nanjing, China
yalinfeng@smail.nju.edu.cn, lyf@nju.edu.cn, mxp@nju.edu.cn

***Abstract*—Although privacy policies serve as the primary mechanism for disclosing data practices under regulations like the General Data Protection Regulation, they frequently contain internal conflicts that undermine transparency and user trust. Existing research has advanced automated privacy policy contradiction analysis by leveraging language models, tuple-based knowledge representations and ontologies (or knowledge graphs) to resolve natural language ambiguities. However, traditional 3-tuple (entity, action, data type) lack contextual information and fail to distinguish data collection practices under varying scenarios, leading to incomplete or misleading contradiction detection. To address these challenges, we present PoliCond, a framework that combines condition-aware tuple representations, domain ontologies and large language models. On established benchmarks, PoliCond achieves an F1 score of 88.6%, outperforming prior methods (58.2%), with an average processing time of 8.4 seconds per policy. In a real-world analysis of 175 privacy policies, PoliCond uncovers previously undetected internal contradictions, including 46 inconsistent policies and 48 contradictory pairs of policy segments missed by existing approaches. These findings underscore the prevalence of inconsistencies in privacy policies and demonstrate the practical utility of PoliCond.**

***Index Terms*—Privacy policy, LLM, Contradiction detection**

I. Introduction

Privacy safety has become an increasingly important and widely discussed issue in recent years. According to the European Data Protection Supervisor annual reports, admissible personal data breach notifications increased from 77 in 2023 [1] to 109 in 2024 [2]. To promote user data protection, a growing number of privacy protection laws have been enacted by various regions and countries, including the General Data Protection Regulation (GDPR) [3], the Data Protection Act (DPA) [4], and the California Consumer Privacy Act (CCPA) [5], which enforce service providers to disclose their privacy policies clearly outlining the collection, usage, and sharing of users' personal data. The introduction of these privacy policies empowers individuals to make informed decisions regarding their personal information and promotes the transparency and accountability of service providers processing such data. Unfortunately, these policies often lack standardization: overly brief ones risk non-compliance by omitting essential disclosures, while verbose policies frequently suffer from internal contradictions, which undermine readability, and could challenge GDPR Article 12 that information should be "transparent, intelligible, and easily accessible".

Recent research has employed natural language processing (NLP) to advance the automated analysis of privacy policies to address such issues [6]. Researchers have focused on generating policy summaries [7], [8] or building question-answering systems [9] for readability, identifying textual contradictions within policies [8], [10] or assessing the consistency between the policies and the de facto functionalities of services provided [11]–[14], evaluating their compliance with relevant privacy regulations [15]–[18], and introducing large language models (LLMs) into this area [19]–[21].

Despite these advances, existing approaches to privacy policy contradiction analysis remain fundamentally limited by their inability to model data collection conditions. Tools like *PolicyLint* [10] typically focus on extracting data collection statements and identify entities, actions, and data types—identifying *who* is involved, *collect* or *not collect*, and *what* information is affected, overlooking the conditions under *which scenarios* the data collection occurs and thus contributing to particular false contradictions. A typical case can be found in Twitter (X.com), "We do not knowingly collect personal information from children under 13" [22]. Existing methods may extract privacy 3-tuples—such as (we, collect, personal information) and (we, not collect, personal information)—without capturing the contextual conditions, leading to a spurious contradiction. Actually, since privacy protection laws like GDPR, DPA, and CCPA propose principles and guidelines for data collection under various scenarios, major organizations frequently issue such conditional clauses in compliance with legal obligations. While *PurPliance* [12] and *PoliGraph* [8] attempt to address this by introducing collection purpose and specialized data types like "personal information @ children", their approaches remain incomplete and fail to account for broader legal scenarios mandated by privacy regulations such as GDPR, DPA, and CCPA. Therefore, modeling a condition-aware tuple format enables explicitly representing the legal and contextual scenarios under which data collection happens.

Note that, including the identification of these conditions in existing analysis frameworks may not be straightforward and could significantly impact precision, as conditions are inherently more challenging to recognize. Unlike entities and data types, conditions are frequently expressed through diverse syntactic structures, contextual dependency and legal con-

∗ Corresponding authors.

ventions, which usually cross sentence boundaries, requiring models to capture long-context and semantic dependencies.

To address these challenges, we propose POLICOND, a framework integrating conditions into traditional 3-tuples to detect contradictions within privacy policies. POLICOND employs a 4-tuple format: (Entity, Action, DataType, Condition-Expr). To model conditions, we build a Condition Ontology. Its conceptual structure derived from OPP-115 [23], while its textual detection patterns are mined across 15,013 real privacy policies, used to standardize common collection conditions. Additionally, POLICOND introduces a pipeline that combines ontologies with large language model's (LLM) reasoning ability. It first uses ontologies to provide candidate values during prompting, then leverages ontologies to validate and normalize outputs, aiming at mitigating LLM hallucinations and enhancing alignment. We propose corresponding condition-aware contradiction rules to identify contradictions supported by the Condition Ontology.

Experimental results on the *PoliGraph-Manual-Correction-Relations* dataset [8] highlight the remarkable performance of POLICOND, which achieves 88.6% F1 score for tuple extraction, with an average processing time of 8.4 seconds per policy. This represents an obvious improvement over PoliGraph, the current state-of-the-art (SOTA) method, which achieves 58.2% F1 score on the same benchmark. Furthermore, the analysis on 175 contemporary privacy policies reveals that 46 policies contain verified contradictions, underscoring its practical utility in auditing the privacy policies' internal consistency.

In this paper, we make the following contributions:

- We propose a novel **Condition Ontology** for modeling data collection contexts and legal alignment, where condition concepts derive from OPP-115 and concept detection patterns mined from 15,013 policies.
- We propose a condition-aware 4-tuple as an information representation, while leveraging the integration of LLMs and ontologies for tuple extraction.
- POLICOND introduces novel condition-aware contradiction rules and evaluates them on 175 real-world apps, revealing 46 contradictory privacy policies, including 48 previously undetected contradictory pairs of policy segments.

The remainder of this paper proceeds as follows: Section 2 reviews the background. Section 3 presents detailed design. Section 4 evaluates our approach. Section 5 describes related work. Section 6 concludes.

II. BACKGROUND

A. Privacy Policy Contradiction

The absence of standardized frameworks for privacy policies often results in contradictions—internal inconsistencies where a policy simultaneously affirms and denies the same or related data practices, particularly around data collection and protection commitments. PolicyLint [10] first formalized this notion, showing such contradictions weaken user trust and undermine credibility. Detecting them is a prerequisite for reliable downstream analysis: unaddressed contradictions cause inconsistent or erroneous results in tasks such as policy summarization, compliance auditing, or behavior-policy alignments. Before any further interpretation or analysis can occur, privacy policies must be internally consistent. Our work builds directly on this foundation, focusing specifically on contradictions in data collection statements.

B. Ontology

Ontologies provide a formal specification of domain-specific concepts and their semantic relations, including *subsumption* (the *is-a* relation, denoted $A \sqsubseteq B$), *part-of*, and *synonymy*, serving as powerful tools to resolve the ambiguity in privacy policy analysis [9], [10], [24]. Although ontologies are commonly represented as directed labeled graphs, the meaning of nodes varies across different abstraction levels: in some ontologies, nodes represent lexical terms; in ours, each node corresponds to a *concept* that denotes a set of synonymous words or phrases. For instance, phrases such as *email address* and *e-mail* are unified under the concept *email*, represented as a node in the graph. Moreover, the concept *personal information* subsumes the concept *email*, which corresponds to a directed *is-a* edge from node *email* to node *personal information*. POLICOND leverages domain-specific ontologies as external knowledge to guide LLMs in generating unambiguous and contextually appropriate outputs, thereby supporting the automated analysis for privacy policies. POLICOND uses four ontologies, covering entities, actions, data types, and conditions.

III. POLICOND

Figure 1 presents the overview of POLICOND. After extracting text, it first splits the privacy policy into individual statements and employs ontologies and Named Entity Recognition (NER) techniques to extract candidate features, resulting in a set of *Collection Statements* and associated candidates. These statements, enriched with more contexts, are then combined with a *Prompt Template* to invoke LLMs. LLMs' outputs undergo post-processing to produce 4-tuples $(Entity, Action, DataType, ConditionExpr)$. Eventually, the framework applies contradiction detection rules to identify potential internal contradictions based on 4-tuples.

A. Definition

Given a privacy policy $\mathcal{P}$, we first divide $\mathcal{P}$ into individual sentences and process them sequentially. We define the set of sentences extracted from a privacy policy as $\mathcal{S}$. POLICOND represents *data collection practices* as a set of four-tuples, denoted as $\mathcal{T}$. Specifically, each tuple $\tau \in \mathcal{T}$ is defined as $\tau = (Entity, Action, DataType, ConditionExpr)$, where *Entity* represents the legal entity performing the data operation, *Action* indicates whether data is collected or not, *Data Type* refers to the collected user data, and *ConditionExpr* represents the collection conditions. Thus, POLICOND is formulated as a mapping function that generates tuples from a given sentence,

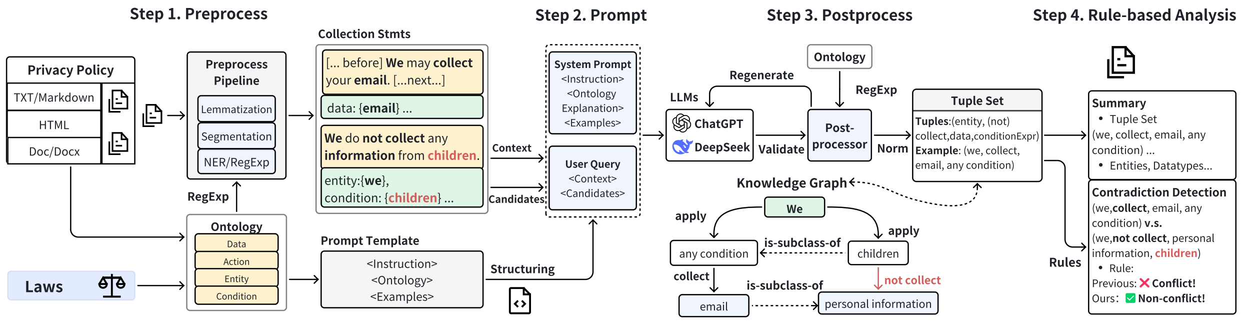


Fig. 1: Overview of PoliCond

denoted as $\psi : \mathcal{S} \to 2^{\mathcal{T}}$. The function ψ represents a one-to-many correspondence, as a single statement can generate multiple tuples. Particularly, for $s \in \mathcal{S}, \mathcal{T}_S \subseteq \mathcal{T}, \psi(s) = \mathcal{T}_S$, we call statement s an **Source** of each tuple $\tau \in \mathcal{T}_S$.

Except condition, other three fields are restricted to corresponding ontologies. We define $ConditionExpr$ as:

$$\begin{aligned} \text{ConditionExpr} \to\ & \text{ConditionExpr}\ \texttt{and}\ \text{ConditionExpr} \\ \mid\ & c\ \texttt{and}\ \text{ConditionExpr} \\ \mid\ & c \end{aligned}$$

where c is a condition item from the Condition Ontology.

This definition allows it to represent compound conditions (e.g., *parental consent* involves both *children* and *user consent* condition). Although both *and* and *or* are semantically valid for composing compound conditions, we apply the following normalization rules to simplify them:

Rule 1 **Disjunction Normalization:** A condition with an *or* relation is normalized by:

$$(e, a, d, c_1\ \texttt{or}\ c_2) \equiv \begin{cases} (e, a, d, c_2), & \text{if } c_1 \sqsubseteq c_2 \\ (e, a, d, c_1), & \text{if } c_2 \sqsubseteq c_1 \\ (e, a, d, c_1),\ (e, a, d, c_2), & \text{otherwise} \end{cases}$$

Rule 2 **Conjunction Subsumption**: A condition combining two condition items via *and* is always more specific than either of its components:

$$(c_1\ \texttt{and}\ c_2) \sqsubseteq c_1 \quad \text{and} \quad (c_1\ \texttt{and}\ c_2) \sqsubseteq c_2$$

Rule 3 **Conjunction Normalization**: When one condition subsumes another, the conjunction simplifies to the more specific condition:

$$(e, a, d, c_1\ \texttt{and}\ c_2) \equiv \begin{cases} (e, a, d, c_1), & \text{if } c_1 \sqsubseteq c_2 \\ (e, a, d, c_2), & \text{if } c_2 \sqsubseteq c_1 \end{cases}$$

where e, a, d denote items from their corresponding ontologies and c is a ConditionExpr.

Based on the 4-tuples, a candidate **Contradiction** in PoliCond is a pair of tuples (τ_c, τ_n), where the positive one describes a specific data collection practice and the other makes a broad, unconditional promise not to collect related data — yet both refer to semantically related entities, data types, or conditions. This signals an internal inconsistency: the policy simultaneously affirms and denies comparable data collection under overlapping contexts, weakening user trust. A **Narrowing**, by contrast, occurs when a broad data collection statement is later compared with a specific exception. For instance, claiming to collect "personal information" while explicitly promising not to collect "email addresses" is a reasonable data practice. Clearly, *contradictions* pose a more severe threat to policy credibility than *narrowings*.

In comparison to traditional $(Entity, Action, DataType)$ 3-tuples, 4-tuples retain more information, such as:

Example III.1. For statement $s_1 \in \mathcal{S}$, "the privacy owner will not collect users' personal information if users are under 13", it can be represented by the tuple $\tau_1 \in \mathcal{T}, \tau_1 =$ (We, not collect, personal information, children).

Example III.2. For statement $s_2 \in \mathcal{S}$, "the privacy owner will collect users' personal information", it can be represented by the tuple $\tau_2 \in \mathcal{T}, \tau_2 =$ (We, collect, personal information, any condition).

As the examples show, $\tau_1.ConditionExpr = children$ encodes a contextual constraint: data collection is explicitly withheld only for users under 13. In contrast, τ_2 makes a general claim with a placeholder condition *any condition*. If processed by prior work [10], which ignores contextual conditions, these two statements would be misclassified as a **contradiction**, despite being logically compatible. PoliCond 's condition-aware design aims to correctly interpret this as a *narrowing*. In total, PoliCond extracts user data collection information from privacy policies to ontology-constrained 4-tuples and identify contradictions based on them.

B. Ontologies

To address linguistic variability in privacy policies, PoliCond extends PoliGraph's ontology framework [8] in two key ways: (1) it introduces a new *Condition* ontology, complementing original *Entity*, *Action*, and *Data* ontologies; (2) based on empirical analysis of real-world policies, it further extends the *Entity* and *Data* ontologies to better capture more practical patterns for existing concepts. Our analysis draws upon 15,013 documents from multiple datasets, including: OPP-115 [23], the Princeton dataset [25], DMOZ [26], PoliCheck [8], [11], a 2024 policy corpus [27], and 175 policies crawled according to AppFigures' rankings [1] .

Ontology Creation. Creating a credible Condition Ontology that aligns with legal terminology and other three existing ontologies to form a comprehensive unified ontology is a non-trivial task. The key insight behind our POLICOND's ontology creation methodology is that "conditions of similar classification typically convey consistent semantic meanings". Our analysis of 28,047 age-related data collection statements, extracted from the aforementioned datasets, indicates that 26,110 (93.1%) explicitly restrict the collection of children's data, typically employing terms such as "under 13", "under 16", or "children" [2]. This convergence may be attributed to companies' efforts to ensure compliance with privacy protection laws and regulatory guidelines that encourage standardized phrasing and condition structures. Building upon this finding, POLICOND adopts a lexicon-guided creation process for the ontology, in which three lexicons, not only for conditions, but also for entities and data types, are collected that map their diverse expressions to shared, canonical concepts in privacy policies. Afterwards, conditions, entities, and data types in privacy policies are then normalized against this lexicon, serving as the foundation of our ontology.

Figure 2 provides a detailed depiction of this process, in which the three Lexicons and Entity, Condition, and Data type Ontologies are created in a semi-automated and LLM-assisted manner. The process begins by extracting data collection behaviors from privacy policies of famous benchmark OPP-115 [23] into the form of four-tuples. Specifically, LLMs are prompted with the instruction: "Extract data collection information from the given text of a privacy policy into a 4-tuple structure: (entity, action, data, condition)...". Note that, the generated results are captured as free text. Tuples with meaningless tags such as 'Unspecified Data' or with illegal format are excluded for uselessness in modeling. Subsequently, we perform an automated lemmatization using the spaCy model *en_core_web_lg* [3] to reduce redundancy, which is a widely used library supporting most of NLP tasks.

The third step is the most critical, wherein entities, datatype, and condition values are extracted from the four-tuples and treated as domain-specific corpora. Part-of-speech tagging with *en_core_web_lg* and TF-IDF are employed to identify linguistically meaningful units—such as nouns with a high TF-IDF rank (e.g., advertising) will be considered candidate concepts. Next, these candidate lexicons are selected based on semantic alignment with OPP-115 annotations [23] or GDPR clauses [3], like those concerning age restrictions (children), user rights and consent requirements (user consent), and common purposes (advertising), etc. Furthermore, MAPP [28], one of the latest privacy policy corpus targeting mobile apps with 64 privacy policies in English and 91 privacy policies in German, is included as an additional reference to ensure broader coverage of emerging privacy practices and to enhance the robustness of candidate concept identification. As no existing condition ontologies are available, we constructed the candidate condition lexicons from scratch. For entity and datatype lexicons, we prioritized reusing candidates from the entity and datatype ontologies of PoliGraph [8], and included others only after careful review and discussion. Three master's students majoring in software engineering and one master's student in law were recruited to perform the task manually. All four students completed at least 8 hours of preparatory study prior to selection, comprising 4 hours of study on the GDPR and 4 hours on the 64 English policies and corresponding data practice annotations in the MAPP corpus. A term was retained as a candidate only when unanimous agreement was reached.

Afterwards, the lexicon is further enriched with semantic relations, including synonymy, hypernymy (or *is-a* relation), and meronymy (*part-of* relation), through a combined LLM-assisted and human-reviewed process. For each condition term, LLMs are prompted to propose candidate relations, which are subsequently reviewed by the students involved in the previous step, with the GDPR serving as an additional reference. For example, phrases like "under 13" are considered a synonym expression of the concept *children*, as students achieve the agreement since GDPR Article 8 proposes "under 13" or "at least under 16 years old" as thresholds for children [3].

Our ontologies' mapping is supported by regex patterns. Eventually, we use the datasets mentioned above to supplement the ontologies' regex patterns and cover more phrasing variations, enhancing its practical applicability.

Condition Ontology. As illustrated in Figure 3, our Condition Ontology focuses on data collection conditions, organized into a hierarchical framework with *subsumption* (*is-a*), *equivalence*, and *part-of* relations. At the root level, *DataCollectionCondition* serves as the top concept, encompassing any possible condition. It branches into the following categories: *ThirdPartyCondition*, *SpecificAudienceCondition*, *ExternalRequirement*, *UserActionCondition*, *PurposeCommitment*, and *DataProtectionCondition*. For simplicity, we take *SpecificAudienceCondition* as an example; it captures restrictions related to user features, such as *Children* and *RegionalUser*. Phrases like "under 13" or "children" will be mapped to the concept *Children*, while phrases like "Californian residents" will be mapped to *RegionalUser*. Notably, the Condition Ontology excludes certain elements (e.g., *Thing*) required for more strict ontology standards (like *Semantic Web*) because they are not necessary at our abstraction level. For brevity, we use abbreviations for conditions throughout the rest of this paper, like *any condition* short for *DataCollectionCondition*.

Other Ontologies. During the ontology creation process, the statistics we gathered also informed our extensions to the existing entity and data-type ontologies. Note that, due to the rapid evolution of privacy policies, capturing all potential entities, datatypes, and other relevant terms when constructing a fully comprehensive ontology is difficult. Nevertheless, our

[1]https://appfigures.com/.

[2]The statements are available in the artifact repository. https://github.com/SEG-DENSE/PoliCond/tree/main/datasets/collectionStmt.

[3]https://spacy.io/models/en.

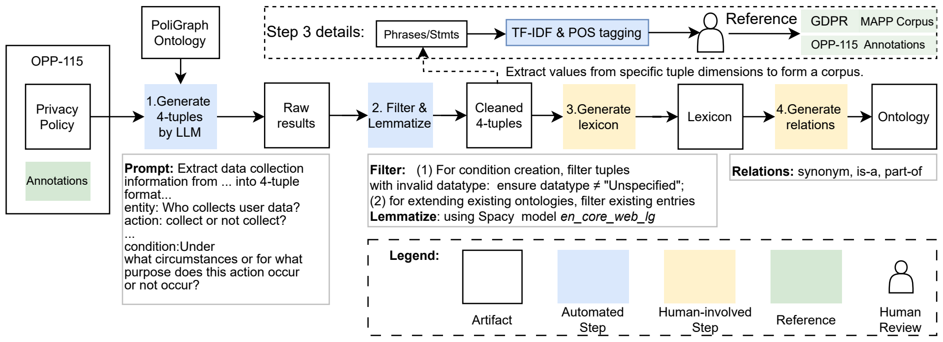

OPP-115
Privacy Policy
Annotations
PoliGraph Ontology
1.Generate 4-tuples by LLM
Raw results
2. Filter & Lemmatize
Cleaned 4-tuples
3.Generate lexicon
Lexicon
4.Generate relations
Ontology
Step 3 details:
Phrases/Stmts
TF-IDF & POS tagging
Reference
GDPR
MAPP Corpus
OPP-115 Annotations
Extract values from specific tuple dimensions to form a corpus.
Prompt: Extract data collection information from ... into 4-tuple format...
entity: Who collects user data?
action: collect or not collect?
...
condition:Under what circumstances or for what purpose does this action occur or not occur?
Filter: (1) For condition creation, filter tuples with invalid datatype: ensure datatype ≠ "Unspecified"; (2) for extending existing ontologies, filter existing entries
Lemmatize: using Spacy model en_core_web_lg
Relations: synonym, is-a, part-of
Legend:
Artifact
Automated Step
Human-involved Step
Reference
Human Review

Fig. 2: Ontology Creation Process

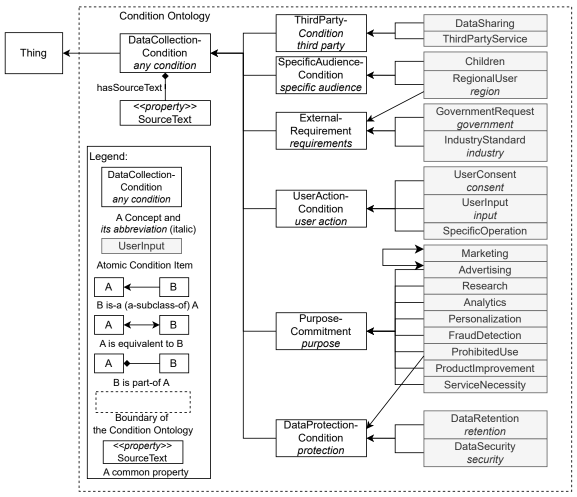


Fig. 3: Condition Ontology for Contradiction Detection

extensions to existing ontologies represent a significant step toward enhancing their overall comprehensiveness.

In the Entity Ontology, prior approaches conflated subsidiaries of specific mega-company into a single entity, leading to overestimation in third-party role categorization. For instance, in the 15,013 documents, Google-affiliated services such as Firebase, Gmail, Google Ads, Google Maps, and Google Play demonstrate coverage frequencies of 372, 725, 1,002, 381, and 751 respectively. If these services are collectively treated as Google and Google is categorized as an *Advertiser*, then aggregating them under *Advertiser* leads to severe overestimation. Our Entity Ontology addresses this by distinguishing subsidiaries and their functional roles: the root node Third Parties branches into mega-companies (e.g., Google) and functional categories (e.g., Advertiser), where subsidiaries maintain many-to-many relations (e.g., Google Ads is simultaneously categorized under both *Google* and *Advertiser*).

For the Data Ontology, we consider location-based recommendation and advertising services as prevalent scenarios, widely described in privacy policies. These services rely heavily on fine-grained user data, where previous Data Ontology lacks such fine-grained concepts. Therefore, we extend the Data Ontology to better capture location-related concepts and relations in these scenarios. For instance, we added a data item *country* after observing its presence in 5,499 documents out of 15,013 (36.6% coverage); added the relationship that *country* is-a *coarse_location* [4]. To align with prior methods [8], [10] and focus on detecting contradictions about user data collection, our Action Ontology only includes two concepts: *collect* and *not collect*, while each concept contains diverse words or expressions as its synonyms.

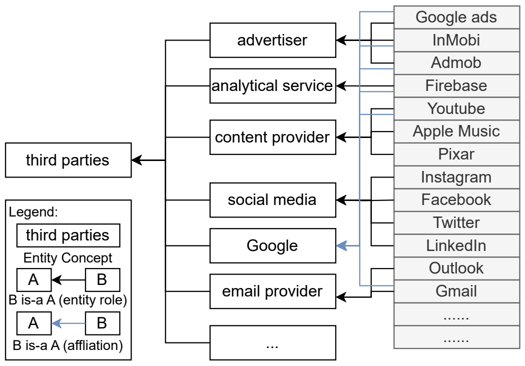


Fig. 4: Entity Ontology

C. Preprocessing

The preprocessing phase is essential for extracting meaningful semantic information and identifying collection statements from privacy policies, which may be published in various formats. The initial step involves extracting textual content and removing HTML tags if the policy is in HTML format. PoliCond then utilizes the spaCy model *en_core_web_lg* to split the content into sentences. After segmentation, PoliCond employs lemmatization to reduce words to their base

[4]Due to space limitations, the figure of our Data Ontology is available in the artifact repository at https://github.com/SEG-DENSE/PoliCond.

forms (e.g., *running* $\rightarrow$ *run*), facilitating Ontology mapping. Next, PoliCond loads regular expressions to match candidate features in the lemmatized sentences. Candidate entities, data types, conditions, and actions will be extracted from sentences. For entities not covered by ontologies' mapping, we also enable using external NER models to recognize entities.

To identify sentences describing data collection practices, we define **Collection Statement** as any sentence in policy P that explicitly or implicitly references the gathering, storing, sharing, or processing of user data. We categorize them into:

1) A *Positive Statement* is a sentence about data collection practices, which requires: (i) a verb indicating data action (e.g., "collect", "use") and (ii) any Data Ontology items;
2) A *Negative Statement* is a sentence indicating a commitment not to collect certain types of user data, which requires: (i) a negation verb phrase (e.g., "not collect", "never use") acting on (ii) any Data Ontology items;

For collection statements, PoliCond dynamically extends their contextual scope by considering two scenarios. Sentences containing sufficient features (entities, data types, or conditions) are treated as self-contained contexts, as exemplified by the statement "We do not knowingly collect personal information from children". In contrast, sentences lacking such features inherit context from the preceding *max_before* sentences, where we search for candidate features in the prior text. Notably, these filtered statements may not represent data collection practices. During preprocessing, our priority is to avoid missing any potential data collection statements, while LLMs are used for semantic understanding.

D. Prompt Process

1) Prompt Rules: Our prompt design enforces 5 key rules: (1) outputs must follow format requirements: LLMs generate *not a collection* for irrelevant sentences, and tuple format *(?, ?, ?, ?)* for collection statements. (2) we prioritize classification over generation, restricting actions to *collect* or *not collect*, and prioritize candidates extracted before. (3) *1-to-n mapping*: single prompt can generate 1-to-n tuples. (4) *no conflict & self-consistency*: each prompt originates from one collection statement, therefore, direct conflict tuples in the same response are discarded. (5) provide explicit Condition Ontology explanation for alignment, allow logical combinations (e.g., *third party and user consent*), and prioritize specific items.

2) Prompt Design: Figure 5 presents the structure of the system prompt and query components, using an example from the privacy policy of Zum Services [29]. The system prompt includes three key components: **Instruction**, **Rules**, and **Examples**, while the query includes the context and candidate information. For the example setup, we adopt a few-shot learning approach [30], providing 4 representative cases: (1) fixed response for irrelevant scenarios, answering 'Not a collection'; (2) a tuple (we, collect, location, any condition); (3) dual tuples for a given (Entity, Data Type) pair, which includes a *collect* tuple and a *not collect* tuple with different and non-conflict conditions; (4) a tuple with combined conditions like *data security and third party*.

E. Post-processing

After prompting, the post-processing stage ensures the reliability of LLM outputs mainly through 3 key steps: validation, regeneration, and normalization.

During validation, PoliCond checks whether the outputs are returned in time and follow the expected format specified in the prompt. PoliCond categorizes the outputs into: (1) *invalid* (e.g., request failures or format violations), which trigger regeneration if the number of retries is below the *max_retry* threshold; or (2) *valid* or *partially valid* results (e.g., inaccurate term usage) requiring normalization.

In normalization, PoliCond uses ontologies to map linguistic expressions to standardized ontology concepts. For instance, the phrase *location data* is normalized to the canonical concept *location*. If no match is found, placeholders such as *unspecified_entity* and *unspecified_data* are used. Additionally, the rules defined in Section III-A are applied to simplify the results; for example, the conditionExpr *user consent and user action* is unified as *user consent*.

During regeneration, PoliCond reissues the same prompt. But for candidate data types that persistently fail beyond *max_retry* retries, PoliCond generates coarse-grained tuples using placeholders. For example, if a prompt includes the candidate features *email* and the action *collect*, PoliCond adds a placeholder tuple to the results (*unspecified_entity, collect, email, any condition*), ensuring extraction of necessary information in worst-case scenarios.

Additionally, before ending post-processing, it is optional to leverage ontological subsumption relations to infer additional tuples and enhance recall for data collection practices. Specifically, given an existing tuple *(entity, collect, d_1, any condition)* and $d_1 \sqsubseteq d_2$, PoliCond adds *(entity, collect, d_2, any condition)* to the result set. However, such inference is restricted to *collect* actions with non-compound *conditionExpr* tuples to avoid overgeneralization in complex or negation scenarios. It is reasonable to say that if a company collects *name* and *email*, then it also collects *personal information*; however, under negation (e.g., "does not collect email") or compound conditions, such generalization may be invalid.

IV. Application

Before formally defining contradictions, we start with such operators:

$$\text{IsRelated}(a,b) = \begin{cases} \text{True}, & \text{if } a \sqsubseteq b \vee b \sqsubseteq a; \\ \text{False}, & \text{otherwise}; \end{cases}$$

where a and b are concepts from the same ontology. We now define the semantic relatedness between two tuples t_1 and t_2 as follows:

$$\text{IsRelatedTuple}(t_1,t_2) = \bigwedge_{\text{dim}\in\{\text{entity},\text{datatype},\text{condition}\}} \text{IsRelated}(t_1.\text{dim}, t_2.\text{dim})$$

Contradictions and narrowings only emerge between related tuples with opposite actions.

As Table I demonstrates, our framework defines four conditional rules based on entity, data, and condition relationships.

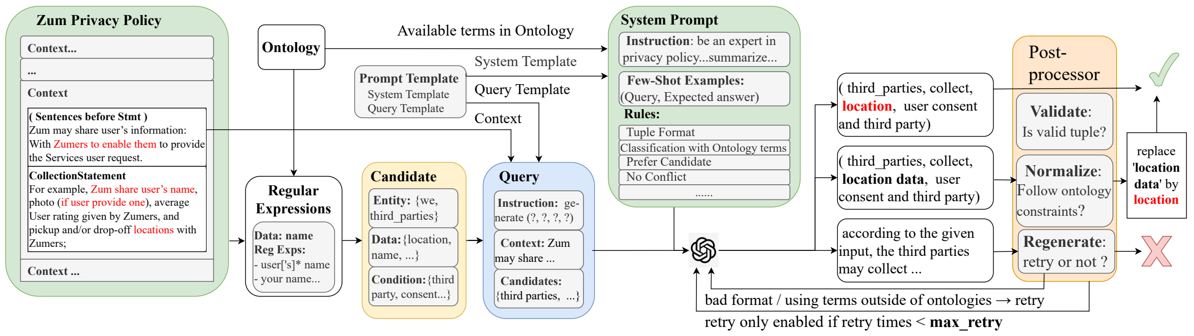

Zum Privacy Policy
Context...
...
Context
(Sentences before Stmt)
Zum may share user's information: With Zumers to enable them to provide the Services user request.
CollectionStatement
For example, Zum share user's name, photo (if user provide one), average User rating given by Zumers, and pickup and/or drop-off locations with Zumers;
Context ...
Ontology
Available terms in Ontology
Prompt Template
System Template
Query Template
System Template
Query Template
Context
Regular Expressions
Data: name
Reg Exps:
- user['s]* name
- your name...
Candidate
Entity: {we, third_parties}
Data: {location, name, ...}
Condition: {third party, consent...}
Query
Instruction: generate (?, ?, ?, ?)
Context: Zum may share ...
Candidates: {third parties, ...}
System Prompt
Instruction: be an expert in privacy policy...summarize...
Few-Shot Examples: (Query, Expected answer)
Rules:
Tuple Format
Classification with Ontology terms
Prefer Candidate
No Conflict
......
(third_parties, collect, location, user consent and third party)
(third_parties, collect, location data, user consent and third party)
according to the given input, the third parties may collect ...
Post-processor
Validate: Is valid tuple?
Normalize: Follow ontology constraints?
Regenerate: retry or not ?
replace 'location data' by location
bad format / using terms outside of ontologies → retry
retry only enabled if retry times < max_retry

Fig. 5: Prompt Design Example for Züm Privacy Policy [29]

TABLE I: Conditional Contradiction Rule Definitions

Contradiction: Collection with Specific Condition
Premise: $(condition_c \sqsubset condition_n) \land IsRelated(entity_c, entity_n) \land IsRelated(data_c, data_n)$ Example: (third parties, collect, personal info, user consent) vs. (google, not collect, email, any condition)
Contradiction: Identical Condition and Specific Collection
Premise: $((condition_c \equiv condition_n) \land (data_c \sqsubset data_n)) \lor ((condition_c \equiv condition_n) \land (data_c \equiv data_n) \land (entity_c \sqsubseteq entity_n))$ Example: 1. (companyX, collect, email, condition) vs. (companyX, not collect, personal info, condition) 2. (companyX, collect, email, condition) vs. (third parties, not collect, email, condition)
Narrowing: Collection with Generic Condition
Premise: $(condition_c \sqsupset condition_n) \land IsRelated(entity_c, entity_n) \land IsRelated(data_c, data_n)$ Example: (advertiser, collect, location, any condition) vs. (google ads, not collect, personal info, children)
Narrowing: Identical Condition and Generic Collection
Premise: $((condition_c \equiv condition_n) \land (data_c \sqsupset data_n)) \lor ((condition_c \equiv condition_n) \land (data_c \equiv data_n) \land (entity_c \sqsupset entity_n))$ Example: 1. (google, collect, personal info, any condition) vs. (google, not collect, email , any condition) 2. (advertiser, collect, email, any condition) vs. (google ads, not collect, email, any condition)

Dual tuples with related values produce narrowing when negation specifies precise scenarios (special cases), while contradictions emerge from vague or impractical commitments. The design emphasizes the necessity for non-collection commitments to be precise and clear, thereby preventing ambiguous, risky or unpractical interpretations. When comparing under identical conditions, our rules build upon PolicyLint's [10] core principle, that a specific data collection claim contradicts a generic negation. When determining which tuple is more specific, PoliCond evaluates specificity based on the relative importance as: condition $>$ data type $>$ entity.

V. Evaluation

In this section, we evaluate PoliCond by answering the following research questions (RQs):

- **RQ1: Effectiveness of the Methodology.** How well does PoliCond extract data collection information from OPP-115 compared to ontology-based and LLM-based baselines?
- **RQ2: Generalization and Comparison.** How well does PoliCond generalize to unseen policies, and how does its performance and efficiency compare against SOTA?
- **RQ3: Application.** What practical value does PoliCond demonstrate in detecting contradictions in privacy policies?

A. Experimental Setting

PoliCond accessed LLM APIs via an asynchronous client, including *OpenAI*, *DeepSeek*, and *Llama*, with a low temperature set to 0.2, following the common practice for relatively stable results. The max_tokens is set to 512 to balance cost and effectiveness. The max_before is set to 3, sufficient to capture relevant context, while avoiding introducing much noise into the context. For preprocessing, we utilize spaCy's model *en_core_web_lg* version 3.7.1 for segmentation and lemmatization. For postprocessing, we set failure-retry limit to 3 also for balancing cost and effectiveness; we only allow PoliCond to infer additional tuples with *collect* action and *any condition*. Experiments were conducted with a laptop with an AMD 8-core CPU, 16 GB memory, running Windows 11. Three LLMs were selected based on their strong NLP performance and widespread adoption in the research and industry communities: *GPT-4o* [31], [32]; *DeepSeek-V3* [33]; *Llama-3.3-70B-Instruct* [34]. We denote PoliCond integrated with these models as $PoliCond_G$, $PoliCond_D$, and $PoliCond_L$, respectively. Unless explicitly specified, PoliCond represents $PoliCond_D$.

B. RQ1: Effectiveness of the Methodology

For RQ1, we evaluate PoliCond 's effectiveness in extracting structured data collection information against a range of representative baselines on the OPP-115 benchmark. They include: (1) PoliGraph [8], the knowledge-graph-based symbolic method, using a RoBERTa-based pipeline and a knowledge graph composed of three domain ontologies, to capture cross-sentence context and subsumption relationships; (2) PolicyGPT [21], a zero-shot prompting and LLM-based method, designed for classifying privacy practices; (3) Rodriguez's prompt strategy [20], a LLM method that incorporates two-shot learning and carefully engineered prompts to extract data practices. By selecting baselines spanning symbolic, zero-shot, and prompt-engineered methods, we assess POLICOND 's novel combination of LLMs, prompt engineering and domain ontologies.

We focus on two OPP-115 categories about data collection: *First Party Collection/Use* and *Third Party Sharing/Collection*. We randomly sampled 15 files from OPP-115, and selected annotations of the two categories as ground truth. Each annotation includes the entity (e.g., "we" for *First Party Collection/Use* or the attribute *Third Party Entity* for another category) and the data type *Personal Information Type*. Although LLM-based baselines were not originally designed for the extraction task, we adapted their prompting strategies. Evaluation is thus conducted by comparing the set of generated (entity, collect, datatype) 3-tuples against human-annotated ground truth. However, since these LLM outputs are inherently unstructured and unconstrained, they require manual validation or manually crafted rules, a process that unavoidably introduces human subjectivity. We adopt the F1-score of precision and recall to evaluate the effectiveness. Assume that the set of annotated tuples $(entity, collect, datatype)$ is recorded as T. For a given privacy policy segment, let T_m denote the set of $(entity, collect, datatype)$ tuples extracted by a previous *method*. We compute the F-score as follows: $precision = \frac{|T_\mathrm{m} \cap T|}{|T_\mathrm{m}|}$, $recall = \frac{|T_\mathrm{m} \cap T|}{|T|}$, and $\mathit{F1\text{-}score} = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{precision + recall}$.

TABLE II: Information Extraction F1-Score on OPP-115

Tool	**#** T_m	**#** $T_m \cap T$	**% Prec.**	**% Recall**	**% F1**
POLICOND	264	**241**	**91.3%**	**86.4%**	**88.8%**
PoliGraph	303	217	71.6%	77.8%	74.6%
PolicyGPT	359	130	36.2%	46.6%	40.8%
Rodriguez's prompt	104	89	85.6%	31.9%	46.5%
Ground-Truth	279				

Table II summarizes the experimental results and highlights clear differences across various methods. PolicyGPT [21], using zero-shot prompting with no terminology definitions, examples or any other structural constraints, scores lowest with an F1 of 40.8%, which stem from the inherent randomness of LLMs. Rodriguez's method [20] achieves high precision (85.6%) for carefully designed prompt and context. However, it provides the entire privacy policy in a single prompt as contextual information rather than segmenting it. Since each prompt can be considered as a single stochastic sampling from the LLM, this approach often extracts too few data practices, resulting in low recall. Additionally, it is adaptable but not well-suited for this task for ignoring entity roles (like third-party). PoliGraph [8] performs solidly, largely due to its knowledge graph. Its high recall stems from its modeling of subsumption (is-a) relationships. During evaluation, we map its subsumption edges into additional collect tuples: for example, if the graph contains (a, collect, c) and $c \sqsubseteq b$, we derive the inferred tuple (a, collect, b), which allows PoliGraph to reduce omissions by subsumptive relation.

POLICOND outperforms all baselines with an 88.8% F1-score. The performance gain of POLICOND over PolicyGPT [21] and Rodriguez's method [20] reveals that: while carefully engineered prompts can significantly improve precision, they offer no guarantee for recall. In contrast, POLICOND leverages external ontological knowledge to capture candidate data practices, dynamically expands their context, invokes LLMs with various candidates, and normalizes LLMs' outputs, finally ensuring broader and more complete coverage of data collection information, which directly addresses the recall deficiency of pure prompting strategies. The advantage over PoliGraph [8] further confirms the effectiveness of combined LLMs and ontologies. Unlike PoliGraph's RoBERTa-based pipeline, POLICOND benefits from the LLMs' extensive pre-training and stronger semantic understanding, whose capabilities are further amplified by prompt or context engineering.

C. RQ2: Generalization and Comparison.

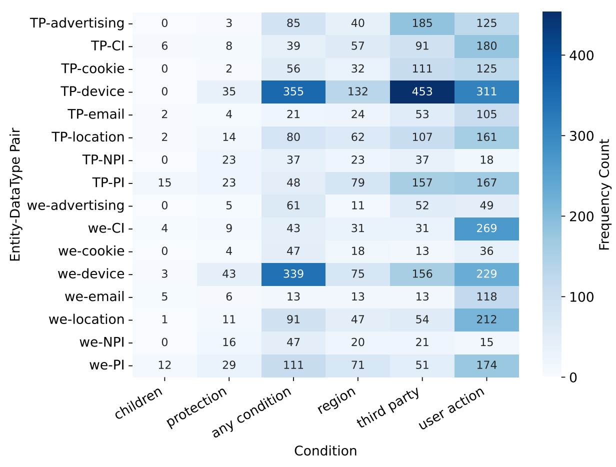
To address RQ2, we evaluated POLICOND on the open dataset PoliGraph-Manual-Correction-Relations [8], comprising 185 policies (2013–2023) with 940 validated 3-tuples.

POLICOND generated 6,230 tuples in total, including 4,803 (77.1%) *collect* tuples, and 1,427 (22.9%) *not collect* tuples. On average, each policy contained 34 tuples, 5 entities, 16 data types, and 12 condition expressions. Among all tuples, 4,692 (75.3%) had atomic conditions, while 1,538 (24.7%) had combined condition expressions. Figures 6a and 6b visualize (entity, data)–condition frequencies, revealing distinct patterns in collect and not collect tuples. For simplicity, we normalized entities to {we, third parties} and preserved following conditions: {children, region, protection, third party, user action, any condition}. Region and children were preserved as atomic conditions while others were consolidated.

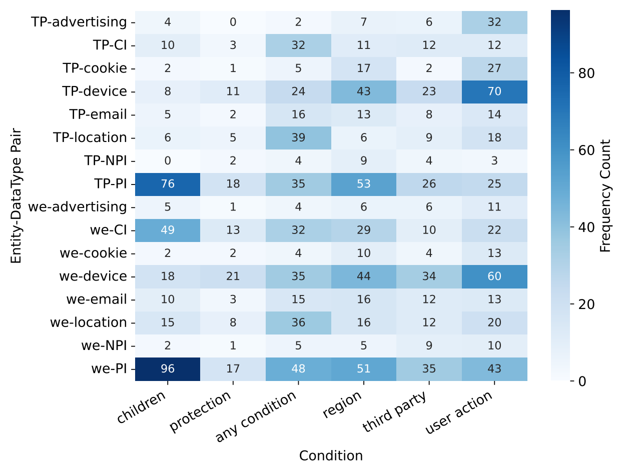
Finding: Children Protection. Our analysis identifies a clear trend of prevalent emphasis on children's data protection among policies, evidenced by the frequency contrast of *collect* tuples compared to *not collect* tuples, demonstrating most policies' compliance with GDPR requirements [3] and confirming our previous findings regarding phrases related to children. Notably, policies tend to use broad terms like **personal information** (PI) when describing children protection. While legally compliant, this generality and ambiguity may obscure what specific data types are restricted, creating potential interpretive gaps for users or auditors.

We proceeded to evaluate the accuracy of generated results through automated comparison and manual verification in the following paragraphs.

Automated Comparison. For automated comparison, we select PoliGraph [8] and its prior work PolicyLint [10] as baselines. LLM-only methods are excluded for inappropriate task and unstructured outputs as described in RQ1. Following PoliGraph's evaluation, we selected the annotated *collect* tuples as ground truth, totaling 878. To bridge semantic gaps between POLICOND's ontology and PoliGraph's schema, we established manually crafted mappings. Generally, our specific concepts are considered equal with PoliGraph's broad concepts, such as *location*, *coarse_location*, and *GPS* equal



(a) Entity-Data-Condition Frequency in Collect Tuples



(b) Entity-Data-Condition Frequency in Not Collect Tuples.

Fig. 6: Statistics of POLICOND's Tuples. PI (Personal Information), CI (Contact Information), NPI (Non-personal Information).

TABLE III: Results on PoliGraph-Manual-Correction-Relations and Ablation Studies

Tool	# T_f	# $T_f \cap T$	% Prec.	% Recall	% F1
$PoliCond_D$	**736**	**715**	97.1%	**81.4%**	**88.6%**
$PoliCond_G$	720	693	96.3%	78.9%	86.7%
$PoliCond_L$	715	696	97.3%	79.3%	87.4%
PoliGraph	640	442	69.1%	50.3%	58.2%
PolicyLint	291	86	29.6%	9.8%	14.7%
Ablation					
$no-post_D$	634	628	**99.1%**	71.5%	83.1%
$no-post_G$	595	582	97.8%	66.3%	79.0%
$no-post_L$	617	609	98.7%	69.4%	81.5%
raw_D	265	259	97.7%	29.5%	45.3%
raw_G	173	164	94.8%	18.7%	31.2%
raw_L	241	234	97.1%	26.7%	41.8%
Ground-Truth	878				

Note: Ablation variants are denoted by: *no-post* disables post-processing, and *raw* disables both post-processing and candidates in prompting.

to PoliGraph's *geographical location*. We follow the F1-score calculation defined in RQ1. Let T denote the ground truth set of (entity, collect, datatype) 3-tuples. Let Outputs(f) denote the set of generated 3-tuples by method f, and T_f denote the subset of Outputs(f) that is relevant to the ground truth. We define T_f by the following filtering rule:

$$T_f \equiv \{t_f \mid t_f \in \text{Outputs}(f) \wedge \exists t_g \in T : \text{IsRelatedTuple}(t_f, t_g)\}$$

This filtering ensures that only related tuples are included for evaluation. We calculate $precision = \frac{|T_f \cap T|}{|T_f|}$, $recall = \frac{|T_f \cap T|}{|T|}$, and *F1-score* as the harmonic mean of precision and recall. As shown in Table III, $PoliCond_D$ achieves a SOTA F1 (88.6%), indicating remarkable improvement over PoliGraph [5].

Ablation Studies. To identify the architectural components driving POLICOND's performance, we conducted ablation studies. Removing *post-processing* from $PoliCond_D$ reduces recall by about 10%. The raw_D variant, which disables both post-processing and ontology-guided prompting, suffers a 42% drop in recall compared to $no-post_D$, demonstrating that both prompt engineering and post-processing enhance alignment and improve recall. We skipped a third variant (post-processing enabled, candidates and preprocessing disabled) as RQ1 shows prompting alone may mislead LLMs to choose incorrect terms.

[5]PoliGraph's F1 is lower than its reported numbers [8], due to differences in evaluation setup.

TABLE IV: Manual Verification of Tuples

Category	# True	# Total	% Accuracy
action = collect	138	173	79.8%
action = not collect	7	15	46.7%
entity = we	94	121	77.7%
entity $\neq$ we	51	67	76.1%
data = email	33	34	97.1%
data = advertising_id	9	12	75.0%
conditionExpr = any condition	37	53	69.8%
conditionExpr $\sqsubseteq$ children	3	4	75.0%
conditionExpr $\sqsubseteq$ region	10	13	76.9%
conditionExpr $\sqsubseteq$ third party	21	31	67.7%
total	145	188	77.1%

Manual Verification. Despite POLICOND 's strong performance on 3-tuples, its ability to correctly generate condition-aware 4-tuples remained unverified. Therefore, we sampled 188 (20%) of the 940 tuples (including *not collect* ones). The author manually verified whether POLICOND's 4-tuples accurately extended corresponding 3-tuples and whether their conditions were grounded in source contexts. As Table IV shows, the overall accuracy was 77.1% (145/188), with significant variation across semantic dimensions. Most importantly, it confirmed a notable deficiency in negation handling (46.7% accuracy). Performance varied by other fields: recognition was highly accurate for *email* but lower for *advertising_id*. This revealed POLICOND's weaknesses in negation reasoning.

Cost. As Table V shows, the efficiency of POLICOND varies across LLM implementations and environments. $PoliCond_G$

TABLE V: Cost on PoliGraph-Manual-Correction-Relations

Model	Time(min)	Tokens	Time(sec)$_{avg}$	Tokens$_{avg}$
$PoliCond_D$	26.0	3602294	8.4	19,472
$PoliCond_G$	14.3	3036619	4.6	16,414
$PoliCond_L$	52.4	3542302	17	19,148
PolicyLint	56.7	–	19.5	–
PoliGraph	13.6	–	4.4	–
Ablation				
raw_D*	288.2 (45.0)	3441861	93.5 (17.9)	18,605
raw_G	39.7	2982243	12.9	16,120
raw_L	52.5	3297449	17.0	17,824

Note: Time with *. When processing raw_D, the DeepSeek API started rate limits. The time inside the parentheses () represents normal response time.

achieves the lowest token usage. Processing time comparisons show PoliGraph handles policies at 4.4 seconds per policy, while $PoliCond_G$ achieving 4.6 seconds. These privacy policies, averaging 3,358 words per document, are processed efficiently. Even the longest policy with 68,457 words was finished in 32.62 seconds by $PoliCond_G$. Our ablation reveals providing candidates will not significantly increase its cost.

D. RQ3: Application

To address RQ3, we evaluated POLICOND's practical utility by analyzing 175 real-world privacy policies crawled from the official websites of apps in 2024, which were chosen according to Appfigures' app rankings. Our sample includes the top 10 most popular apps and each 5 apps from 33 distinct categories (13 of the 175 had incomplete content due to crawling issues). We focus on detecting internal contradictions within these policies and compare POLICOND's findings against existing baselines. As demonstrated in Fig. 7, when analyzing the 2022 version of The League's privacy policy, POLICOND successfully identifies contradictions by applying its rules. The policy simultaneously permits sharing advertising identifiers and location with marketers yet prohibits third party information collection. These contradictory claims create ambiguity regarding data-sharing practices, eroding trust and reliability.

POLICOND produced an average of 36 tuples per policy, totaling 5889 tuples (4697 collect, 1192 not collect). From these, it identified 2,777 candidate contradiction pairs.

TABLE VI: Contradiction Verification on Collected Policies

Tool	# Candidate	# Verified	# True	% Accuracy	# Unique
$PoliCond_D$ (Tuple)	2777	410	125	30.5%	–
$PoliCond_D$ (Context)	1692	250	69	27.6%	48
$PoliCond_D$ (File)	138	95	46	48.4%	36
$PolicyLint$ (Tuple)	148	148	49	33.1%	–
$PolicyLint$ (Context)	104	104	27	26.0%	4
$PolicyLint$ (File)	27	27	15	55.6%	3
$PoliGraph$ (Tuple)	39	39	38	97.4%	–
$PoliGraph$ (Context)	11	11	10	90.9%	0
$PoliGraph$ (File)	7	7	6	85.7%	0

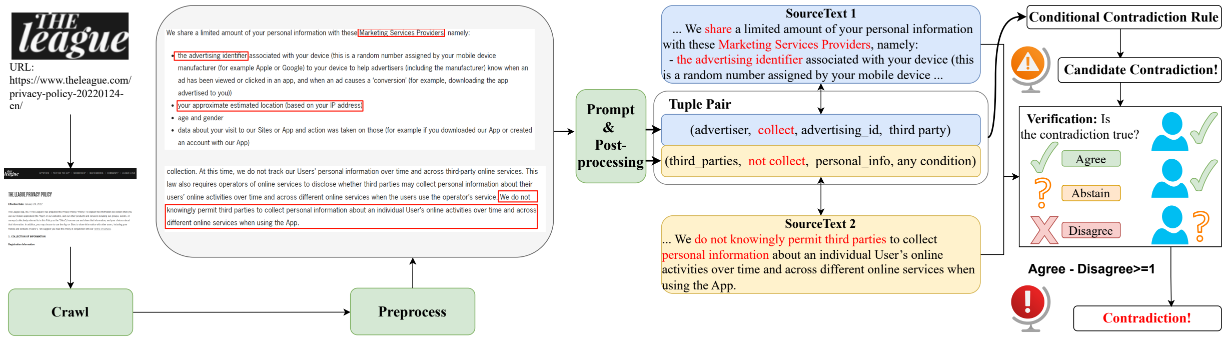
As Table VI shows, contradictions are reported at 3 levels: (1) tuple-level, (2) context-level (each tuple in a contradictory pair is grounded in its own LLM input context, i.e., the set of policy sentences used to generate it), and (3) file-level (a privacy policy, i.e., a policy with verified conflicting tuples is a self-contradictory policy). To enhance reliability, three raters, including the first author and two graduate students in software engineering, performed majority voting on a sample of 250 context-level contradictions. Of these, 69 (27.6%) context pairs were confirmed as contradictory pairs, corresponding to 46 self-contradictory policies and 125 conflicting tuple pairs. The results indicate that internal contradictions are common in real-world policies, primarily arising from overly broad, unqualified non-collection commitments. The results also revealed the advantages and limitations of POLICOND. PolicyLint [10], despite its limited candidates (148 tuples), discovered four context-level contradictions undetected by POLICOND. PoliGraph [8] achieved high accuracy but failed to identify novel contradictions, as it focused on reducing false positives based on PolicyLint. In contrast, POLICOND detected 48 unique context-level contradictions. However, the accuracy of POLICOND is mainly limited by negation accuracy.

VI. THREATS TO VALIDITY

Internal Validity. The primary internal validity threat to our POLICOND lies in its design of capturing conditions and representing them as one element in the 4-tuple, analogous to the treatment of datatypes, actions, and entities. While this design substantially enhances the effectiveness of tuple extraction and the subsequent contradiction analysis, it may miss implicit contradictions that demand more nuanced legal reasoning. As mentioned in Section III-B, conditions in modern privacy policies typically display a high degree of similarity, largely due to their adherence to uniform data protection regulations. Based on this observation, the Condition Ontology in POLICOND is primarily constructed on the basis of OPP-115 [23], with empirical statistics incorporated as an extension. These practices enable POLICOND to identify a broader spectrum of contradictions across a wider range of policy contexts. However, the ontologies may not fully cover emerging practices or conditions in non-English or multi-jurisdictional policies.

External Validity. The generalizability of POLICOND is constrained by the scope of our evaluation corpora. Our ontology and experiments rely heavily on English-language privacy policies, primarily shaped by GDPR, CCPA and DPA, which may not reflect the linguistic complexity, legal frameworks, or structural conventions of policies in other jurisdictions (e.g., non-English or non-Western regulatory contexts). Moreover, among the millions of global services, the 175 real-world policies (analyzed in RQ3) were selected from top-ranked apps, potentially overlooking smaller services.

Limitations. Beyond validity concerns, POLICOND is inherently limited to textual analysis and cannot verify whether stated data practices align with actual app behaviors. Second, the pipeline relies on standard NLP preprocessing (e.g., spaCy segmentation and lemmatization), which may struggle with tables, nested lists, and complex formatting, potentially fragmenting extractions [6], [8]. Ontologies' concept mapping depends on regex patterns, which inevitably miss or wrongly capture specific phrasings. These errors introduced during early stages can propagate through the pipeline. Third, POLICOND inherits LLMs' weakness in negation reasoning [36]–


THE league
URL: https://www.theleague.com/privacy-policy-20220124-en/
Crawl
Preprocess
Prompt & Post-processing
Tuple Pair
(advertiser, collect, advertising_id, third party)
(third_parties, not collect, personal_info, any condition)
SourceText 1
SourceText 2
Conditional Contradiction Rule
Candidate Contradiction!
Verification: Is the contradiction true?
Agree
Abstain
Disagree
Agree - Disagree>=1
Contradiction!

We share a limited amount of your personal information with these Marketing Services Providers namely:

- the advertising identifier associated with your device (this is a random number assigned by your mobile device manufacturer (for example Apple or Google) to your device to help advertisers (including the manufacturer) know when an ad has been viewed or clicked in an app, and when an ad causes a 'conversion' (for example, downloading the app advertised to you))
- your approximate estimated location (based on your IP address)
- age and gender
- data about your visit to our Sites or App and action was taken on those (for example if you downloaded our App or created an account with our App)

collection. At this time, we do not track our Users' personal information over time and across third-party online services. This law also requires operators of online services to disclose whether third parties may collect personal information about their users' online activities over time and across different online services when the users use the operator's service. We do not knowingly permit third parties to collect personal information about an individual User's online activities over time and across different online services when using the App.

... We share a limited amount of your personal information with these Marketing Services Providers, namely:
- the advertising identifier associated with your device (this is a random number assigned by your mobile device ...

... We do not knowingly permit third parties to collect personal information about an individual User's online activities over time and across different online services when using the App.

Fig. 7: Application of POLICOND in Analyzing THE League privacy policy [35]

[38]. Fourth, the contextual information is expanded using a fixed window of preceding sentences, which potentially limits cross-sentence context awareness. Finally, its dependence on external APIs introduces latency, cost, and rate-limit constraints in real-world deployment.

VII. Related Work

Privacy Policy Analysis. Early efforts in this field, like OPP-115 dataset [23], focused on manual annotation and categorization. While foundational, such methods face scalability limitations. Subsequent work leveraged NLP and machine learning: PoliSis [9] applied CNN-based classification to policy segments and introduced question-answering capabilities; PolicyLint [10] pioneered automated extraction of entities and data types to detect policy contradictions; Purpliance [12] advanced purpose extraction; and PoliGraph [8] introduced ontology-based knowledge graphs for cross-sentence context analysis. More recently, PolicyGPT [21] explored zero-shot prompting, while Rodriguez et al. [20] and PAPEL [19] employed prompt engineering to support policy analysis. PRISMe [39] offers an interactive browser extension that leverages LLMs for real-time, question-driven policy assessment. Despite these advances, rule- and graph-based frameworks still lose contextual nuance in data collection scenarios, while LLM-only methods suffer from output instability and reliability concerns.

Requirement Engineering. Requirement engineering provides foundational structure for modeling privacy policies, addressing ambiguity through formal representations. Bhatia and Breaux [40], [41] introduced an information type lexicon derived from manual annotations and further identified semantic incompleteness in privacy policies, demonstrating that the absence of explicit conditions and purposes increases privacy risks. Sleimi et al. [42] leveraged dependency and constituency parsing to extract semantic metadata, establishing a harmonized model that formalizes semantic roles such as conditions and purposes. Hosseini et al. [43] proposed grammar-driven semantic rules to automate ontology construction. Despite these advances, existing methods are limited by manual efforts, rigid grammars, or inability to infer from common knowledge, where LLMs are expected to mitigate such weaknesses.

VIII. Conclusion

This paper introduces a condition ontology for modeling contextual constraints in user data collection, along with a framework that integrates LLMs and ontologies to extract condition-aware data collection tuples, thereby enabling the detection of internal contradictions within privacy policies. Human evaluation shows that POLICOND produces semantically grounded tuples with high accuracy. On established benchmarks, POLICOND outperforms existing methods, achieving an F1 score of 88.6% compared to 58.2%, with an average processing time of 8.4 seconds per policy. In a real-world analysis of 175 privacy policies, POLICOND demonstrates its effectiveness by uncovering previously undetected internal contradictions, including 46 policies with high-risk inconsistencies and 48 contradictory pairs of policy segments missed by prior methods.

IX. Data Availability

Artifacts and datasets of POLICOND are available at https://github.com/SEG-DENSE/PoliCond.

Acknowledgment

We thank the anonymous reviewers for their invaluable feedback on this paper. This research is supported by the National Natural Science Foundation of China under Grant Nos. 62402213 and 62372227.

References

[1] European Data Protection Supervisor, "Edps annual report 2023," European Data Protection Supervisor, Brussels, Belgium, Tech. Rep., 2023, accessed: May 29, 2025. [Online]. Available: https://www.edps.europa.eu/system/files/2024-04/2024-04-09-annual-report-2023_en.pdf
[2] ——, "Edps annual report 2024," European Data Protection Supervisor, Brussels, Belgium, Tech. Rep., 2024, accessed: May 29, 2025. [Online]. Available: https://www.edps.europa.eu/system/files/2025-04/edps_annual_report-2024_en.pdf
[3] European Union, ""General Data Protection Regulation (GDPR)."," *Official Journal L110*, vol. 59, pp. 1–88, 2016-05-04. [Online]. Available: https://gdpr-info.eu/
[4] U. Government, "Data protection act 2018." 2018. [Online]. Available: https://www.legislation.gov.uk/ukpga/2018/12/contents/enacted
[5] State of California Department of Justice, Office of the Attorney General, "California consumer privacy act (ccpa)," 2024. [Online]. Available: https://oag.ca.gov/privacy/ccpa

[6] A. Adhikari, S. Das, and R. Dewri, "Natural Language Processing of Privacy Policies: A Survey," Jan. 2025. [Online]. Available: http://arxiv.org/abs/2501.10319

[7] R. N. Zaeem, R. L. German, and K. S. Barber, "Privacycheck: Automatic summarization of privacy policies using data mining," *ACM Trans. Internet Technol.*, vol. 18, no. 4, Aug. 2018. [Online]. Available: https://doi.org/10.1145/3127519

[8] H. Cui, R. Trimananda, A. Markopoulou, and S. Jordan, "PoliGraph: Automated privacy policy analysis using knowledge graphs," in *32nd USENIX Security Symposium (USENIX Security 23)*. Anaheim, CA: USENIX Association, Aug. 2023, pp. 1037–1054. [Online]. Available: https://www.usenix.org/conference/usenixsecurity23/presentation/cui

[9] H. Harkous, K. Fawaz, R. Lebret, F. Schaub, K. G. Shin, and K. Aberer, "Polisis: Automated analysis and presentation of privacy policies using deep learning," in *27th USENIX Security Symposium (USENIX Security 18)*, 2018, pp. 531–548.

[10] B. Andow, S. Y. Mahmud, W. Wang, J. Whitaker, W. Enck, B. Reaves, K. Singh, and T. Xie, "Policylint: Investigating internal privacy policy contradictions on google play," in *28th USENIX Security Symposium, USENIX Security 2019, Santa Clara, CA, USA, August 14-16, 2019*, N. Heninger and P. Traynor, Eds. USENIX Association, 2019, pp. 585–602. [Online]. Available: https://www.usenix.org/conference/usenixsecurity19/presentation/andow

[11] B. Andow, S. Y. Mahmud, J. Whitaker, W. Enck, B. Reaves, K. Singh, and S. Egelman, "Actions speak louder than words: Entity-sensitive privacy policy and data flow analysis with policheck," in *29th USENIX Security Symposium, USENIX Security 2020, August 12-14, 2020*, S. Capkun and F. Roesner, Eds. USENIX Association, 2020, pp. 985–1002. [Online]. Available: https://www.usenix.org/conference/usenixsecurity20/presentation/andow

[12] D. Bui, Y. Yao, K. G. Shin, J.-M. Choi, and J. Shin, "Consistency Analysis of Data-Usage Purposes in Mobile Apps," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '21. New York, NY, USA: Association for Computing Machinery, Nov. 2021, pp. 2824–2843. [Online]. Available: https://dl.acm.org/doi/10.1145/3460120.3484536

[13] Z. Tan and W. Song, "PTPDroid: Detecting Violated User Privacy Disclosures to Third-Parties of Android Apps," in *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*, May 2023, pp. 473–485. [Online]. Available: https://ieeexplore.ieee.org/document/10172832

[14] K. Zhao, X. Zhan, L. Yu, S. Zhou, H. Zhou, X. Luo, H. Wang, and Y. Liu, "Demystifying Privacy Policy of Third-Party Libraries in Mobile Apps," in *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*, May 2023, pp. 1583–1595. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10172865

[15] T. Nguyen, "An Empirical Evaluation of the Implementation of the California Consumer Privacy Act (CCPA)," Sep. 2022. [Online]. Available: http://arxiv.org/abs/2205.09897

[16] F. Xie, Y. Zhang, C. Yan, S. Li, L. Bu, K. Chen, Z. Huang, and G. Bai, "Scrutinizing Privacy Policy Compliance of Virtual Personal Assistant Apps," in *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering*, ser. ASE '22. New York, NY, USA: Association for Computing Machinery, Jan. 2023, pp. 1–13. [Online]. Available: https://doi.org/10.1145/3551349.3560416

[17] O. A. Cejas, M. I. Azeem, S. Abualhaija, and L. C. Briand, "Nlp-based automated compliance checking of data processing agreements against gdpr," *IEEE Transactions on Software Engineering*, vol. 49, no. 9, pp. 4282–4303, 2023.

[18] S. Pan, D. Zhang, M. Staples, Z. Xing, J. Chen, X. Xu, and T. Hoang, "Is it a trap? a large-scale empirical study and comprehensive assessment of online automated privacy policy generators for mobile apps," in *33rd USENIX Security Symposium (USENIX Security 24)*. Philadelphia, PA: USENIX Association, Aug. 2024, pp. 5681–5698. [Online]. Available: https://www.usenix.org/conference/usenixsecurity24/presentation/pan-shidong-trap

[19] A. Goknil, F. B. Gelderblom, S. Tverdal, S. Tokas, and H. Song, "Privacy policy analysis through prompt engineering for llms," *CoRR*, vol. abs/2409.14879, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2409.14879

[20] D. R. Torrado, I. Yang, J. M. del Álamo, and N. Sadeh, "Large language models: a new approach for privacy policy analysis at scale," *Computing*, vol. 106, no. 12, pp. 3879–3903, 2024. [Online]. Available: https://doi.org/10.1007/s00607-024-01331-9

[21] C. Tang, Z. Liu, C. Ma, Z. Wu, Y. Li, W. Liu, D. Zhu, Q. Li, X. Li, T. Liu, and L. Fan, "Policygpt: Automated analysis of privacy policies with large language models," 2023. [Online]. Available: https://arxiv.org/abs/2309.10238

[22] X. Corp., "Privacy policy," Website, Nov. 2024, accessed: May 27, 2025. [Online]. Available: https://x.com/en/privacy.

[23] S. Wilson, F. Schaub, A. A. Dara, F. Liu, S. Cherivirala, P. Giovanni Leon, M. Schaarup Andersen, S. Zimmeck, K. M. Sathyendra, N. C. Russell, T. B. Norton, E. Hovy, J. Reidenberg, and N. Sadeh, "The Creation and Analysis of a Website Privacy Policy Corpus," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, 2016, pp. 1330–1340. [Online]. Available: http://aclweb.org/anthology/P16-1126

[24] N. Noy, "Ontology development 101: A guide to creating your first ontology," 2001. [Online]. Available: https://api.semanticscholar.org/CorpusID:500106

[25] R. Amos, G. Acar, E. Lucherini, M. Kshirsagar, A. Narayanan, and J. Mayer, "Privacy Policies over Time: Curation and Analysis of a Million-Document Dataset," in *Proceedings of The Web Conference 2021*, ser. WWW '21. Association for Computing Machinery, p. 22. [Online]. Available: https://doi.org/10.1145/3442381.3450048

[26] R. N. Zaeem and K. S. Barber, "A large publicly available corpus of website privacy policies based on DMOZ," in *CODASPY '21: Eleventh ACM Conference on Data and Application Security and Privacy, Virtual Event, USA, April 26-28, 2021*, A. Joshi, B. Carminati, and R. M. Verma, Eds. ACM, 2021, pp. 143–148. [Online]. Available: https://doi.org/10.1145/3422337.3447827

[27] O. A. Cejas, "Web Scraper & Privacy Policies Dataset," https://doi.org/10.6084/m9.figshare.26425306.v1, 2024, accessed:Mar 2025.

[28] S. Arora, H. Hosseini, C. Utz, V. Bannihatti Kumar, T. Dhellemmes, A. Ravichander, P. Story, J. Mangat, R. Chen, M. Degeling, T. Norton, T. Hupperich, S. Wilson, and N. Sadeh, "A tale of two regulatory regimes: Creation and analysis of a bilingual privacy policy corpus," in *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, J. Odijk, and S. Piperidis, Eds. Marseille, France: European Language Resources Association, Jun. 2022, pp. 5460–5472. [Online]. Available: https://aclanthology.org/2022.lrec-1.585/

[29] Zum Services Inc., "Privacy policy," Website, Nov. 2015, archived document from UCI Networking Group's PoliGraph Dataset. The Zum's official website has published a newer version. Accessed: May 28, 2025. [Online]. Available: https://fs.cvhc.cc/poligraph/406739376972e9daf7b781530354d11e678173b3a9b830121e63ba255a167419.html

[30] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, and A. Askell, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., 2020. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html

[31] OpenAI, "GPT-4 technical report," *CoRR*, vol. abs/2303.08774, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2303.08774

[32] A. Hurst, A. Lerer, A. P. Goucher, A. Perelman, A. Ramesh, and A. Clark, "Gpt-4o system card," *CoRR*, vol. abs/2410.21276, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2410.21276

[33] DeepSeek-AI, "Deepseek-v3 technical report," 2025. [Online]. Available: https://arxiv.org/abs/2412.19437

[34] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "Llama: Open and efficient foundation language models," *CoRR*, vol. abs/2302.13971, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2302.13971

[35] The League App, Inc., "Privacy policy," Website, Jan. 2022, effective Date: Jan. 24, 2022. The League's official website has published a newer version. Accessed: May 29, 2025. [Online]. Available: https://www.theleague.com/privacy-policy-20220124-en/

[36] I. García-Ferrero, B. Altuna, J. Álvez, I. Gonzalez-Dios, and G. Rigau, "This is not a dataset: A large negation benchmark to challenge large language models," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023,*

Singapore, December 6-10, 2023, H. Bouamor, J. Pino, and K. Bali, Eds. Association for Computational Linguistics, 2023, pp. 8596–8615. [Online]. Available: https://doi.org/10.18653/v1/2023.emnlp-main.531

[37] M. Ye, T. Kuribayashi, J. Suzuki, G. Kobayashi, and H. Funayama, "Assessing step-by-step reasoning against lexical negation: A case study on syllogism," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, H. Bouamor, J. Pino, and K. Bali, Eds. Association for Computational Linguistics, 2023, pp. 14 753–14 773. [Online]. Available: https://doi.org/10.18653/v1/2023.emnlp-main.912

[38] I. García-Ferrero, B. Altuna, J. Alvez, I. Gonzalez-Dios, and G. Rigau, "This is not a dataset: A large negation benchmark to challenge large language models," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 8596–8615. [Online]. Available: https://aclanthology.org/2023.emnlp-main.531/

[39] V. Freiberger, A. Fleig, and E. Buchmann, ""you don't need a university degree to comprehend data protection this way": Llm-powered interactive privacy policy assessment," in *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, 2025. [Online]. Available: https://doi.org/10.1145/3706599.3719816

[40] J. Bhatia and T. D. Breaux, "Towards an information type lexicon for privacy policies," in *2015 IEEE Eighth International Workshop on Requirements Engineering and Law (RELAW)*, 2015, pp. 19–24. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7330207

[41] ——, "Semantic Incompleteness in Privacy Policy Goals," in *2018 IEEE 26th International Requirements Engineering Conference (RE)*. Banff, AB: IEEE, Aug. 2018, pp. 159–169. [Online]. Available: https://ieeexplore.ieee.org/document/8491132/

[42] A. Sleimi, N. Sannier, M. Sabetzadeh, L. Briand, and J. Dann, "Automated Extraction of Semantic Legal Metadata using Natural Language Processing," in *2018 IEEE 26th International Requirements Engineering Conference (RE)*, Aug. 2018, pp. 124–135, 163. Sleimi ISSN: 2332-6441. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8491129

[43] M. B. Hosseini, T. D. Breaux, R. Slavin, J. Niu, and X. Wang, "Analyzing privacy policies through syntax-driven semantic analysis of information types," *Information and Software Technology*, vol. 138, p. 106608, Oct. 2021. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0950584921000859