An Evaluation of Constituency-based Hyponymy Extraction from Privacy Policies

Morgan C. Evans[1], Jaspreet Bhatia[2], Sudarshan Wadkar[2] and Travis D. Breaux[2]
Bard College, Annandale-on-Hudson, New York, United States[1]
Carnegie Mellon University, Pittsburgh, Pennsylvania, United States[2]
me4582@bard.edu, jbhatia@cs.cmu.edu, swadkar@cs.cmu.edu, breaux@cs.cmu.edu

***Abstract*—Requirements analysts can model regulated data practices to identify and reason about risks of non-compliance. If terminology is inconsistent or ambiguous, however, these models and their conclusions will be unreliable. To study this problem, we investigated an approach to automatically construct an information type ontology by identifying information type hyponymy in privacy policies using Tregex patterns. Tregex is a utility to match regular expressions against constituency parse trees, which are hierarchical expressions of natural language clauses, including noun and verb phrases. We discovered the Tregex patterns by applying content analysis to 30 privacy policies from six domains (shopping, telecommunication, social networks, employment, health, and news.) From this dataset, three semantic and four lexical categories of hyponymy emerged based on category completeness and word-order. Among these, we identified and empirically evaluated 72 Tregex patterns to automate the extraction of hyponyms from privacy policies. The patterns match information type hyponyms with an average precision of 0.72 and recall of 0.74.**

***Index Terms*—Hyponym, hypernym, natural language processing, ontology, privacy policy, compliance.**

I. Introduction

Personal privacy concerns how personal information is collected, used, and shared within information systems. To reduce the risk of privacy violations, regulators require companies to rationalize their data practices and comply with privacy laws, such as the E.U. Data Protection Directive 95/46/EC and the Health Insurance Portability and Accountability Act (HIPAA) Privacy Rule. To help requirements analysts design legally compliant, privacy-preserving systems, new methods have been proposed. For example, Ghanavati's compliance framework for business process specification as applied to Canada's privacy health law [11], Maxwell and Anton's production rule system for extracting legal requirements from privacy law [22], Breaux et al.'s Eddy language for asserting privacy principles [4, 6], and Paja et al.'s STS-ml tool for analyzing privacy and security requirements in socio-technical systems [24]. These methods and the problems that they address present a challenge to requirements analysts: what does the category of personal information formally consist of, in order to infer the consequences of collecting and sharing such information in a formalization of data practices?

In this paper, we report results from developing and evaluating an automated method for extracting information ontology from privacy policies. Privacy policies are posted at most websites, and frequently required by best practice or privacy law, e.g., HIPAA and the Gramm-Leach Bliley Act. These policies describe the data practices of online services, and frequently include data practice descriptions for physical locations where services are rendered. In general, a privacy policy describes what information is collected, how it is used, and with whom it is shared. These descriptions frequently include examples that illustrate relevant kinds of information, called sub-ordinate terminology or *hyponyms*. When interpreting these policies and law, statements that regulate an information type could logically regulate any hyponyms, for example, any restrictions on "contact information" could also be applied to "email address."

The contributions of this paper are four-fold: first, we identify a taxonomy of hyponymy patterns that describes the complete set of hyponyms manually identified among 30 privacy policies; second, we formalize these patterns using Tregex, a tree regular expression language for matching constituency parse trees [18]; third, we report the number of information types covered by these patterns, called coverage, when compared to a lexicon of information types extracted from the same policies using the method by Bhatia et al. [1]; and fourth, we analyze the variation of information types across domains.

The remainder of the paper is organized as follows: in Section II, we review background concepts and related work; in Section III, we present our approach to identifying hyponymy automatically; in Section IV, we present results and the evaluation of our approach; and in Section V, we discuss our results and future work.

II. Background and Related Work

We now review hyponymy in natural language, Tregex and related work.

Hyponyms are specific phrases that are sub-ordinate to another, more general phrase, which is called the *hypernym* [15]. Speakers and readers of natural language typically use the linking verb phrase *is a kind of* to express the relationship between a hyponym and hypernym, e.g., a GPS location is a kind of real-time location. Other semantic relationships of interest include *meronyms*, which describe a part-whole

relationship, *homonyms*, which describe a word that has two unrelated meanings, and *polysemes*, which describe a word with two related meanings [15]. A popular online lexical database that contains hyponyms is called WordNet [23].

Hearst first proposed a set of six lexico-syntactic patterns to identify hyponyms in natural language texts using noun phrases and regular expressions [12]. The patterns are domain independent and include the indicative keywords "such as," "including," and "especially," among others. The Hearst approach applies grammar rules to a unification-based constituent analyzer over part-of-speech (POS) tags to find noun phrases that match the pattern, which are then checked against an early version of WordNet for verification [12]. The approach was unable to work for meronymy in text.

Snow et al. applied WordNet and machine learning to a newswire corpus to identify lexico-syntactic patterns and hyponyms [29]. Their approach includes the six Hearst patterns and resulted in a 54% increase in the number of words over WordNet. Unlike Hearst and Snow et al., information types are rarely found in WordNet: among the 1300 information types used in our approach described herein, only 17% of these phrases appear in WordNet, and only 19% of the phrases matched by our hyponymy patterns appear in WordNet. This means that requirements analysts who want to find the category of an information type, or find the members of an information category, will be unlikely to find these answers in WordNet. Our work aims to identify these hyponyms for reuse by requirements analysts in future projects.

The identification of hyponyms and hypernyms can be considered as a case of categorization phenomena which is studied extensively in cognitive sciences. Of particular interest to our work is Rosch's category theory [25] and Tversky's formal approach to category resemblance [31]. Rosch introduced category theory to define terminology for understanding how abstractions relate to one another [25]. This includes the construction of taxonomies, which relate categories through class inclusion: the more inclusive a category is, the higher that category appears in the taxonomy. Higher-level categories are hypernyms, which contain lower-level categories or hyponyms. In addition, Rosch characterizes categories by the features they share and she uses this designation to introduce the concept of *cue validity*, which is the probability that a cue x is the predictor of a category y. Categories with high cue validity are what Rosch calls *basic-level categories*. In our analysis, hyponyms are frequently linked to a higher-level category that can also be considered as a basic-level category; however, the features that define these categories are not typically found in policy texts, and instead they are tacit knowledge.

An important assumption in Rosch's definition of taxonomy is that each category can at most be a member of one other category. Information type names violate this assumption, because an "e-mail address" can be classified as both "login information" and "contact information," depending on how the e-mail address is used in an information system. Thus, information types may be more amenable to mathematical comparison using Tversky's *category resemblance*, which is a measure in which disjoint categories combine when their shared features outweigh their unshared features [31]. Category resemblance also accounts for asymmetry in similarity [31], which may account for differences arising from confusion among hyponymy, meronymy, homonymy and polysemy. Our approach to extract hyponyms from text does not account for these measured interpretations by Rosch and Tversky, but instead relies on the policy author's authority to control meaning.

In our approach, we use Tregex, which is a utility developed by Levy and Andrew to match constituency parse trees [18]. Constituency parse trees are constructed automatically from POS-tagged sentences in which each word is tagged with a POS tag, such as a noun, verb, adjective, or preposition tag, among others using Stanford CoreNLP [21]. Tregex has been used to generate questions from declarative sentences [13], to evaluate text summarization [30], to characterize temporal requirements [19], and to generate an interpretative regulatory policy [16].

While natural language processing (NLP) of requirements texts can scale analysis to large corpora, the role of NLP should not be overstated, since it does not account for human interpretation [26]. Jackson and Zave argue that requirements engineering is principally concerned with writing accurate software specifications, which require explicit statements about domain phenomena [14]. While significant work has been done to improve specification, a continuing weakness is that problems are frequently formalized using low-level programming concepts (classes, data, and operations) as opposed to using richer, problem-oriented ontologies [17]. In this paper, we investigate an approach for extracting an information type ontology from higher-order descriptions of information systems embodied in privacy policies. We believe these ontologies can improve how we reason about and analyze privacy requirements for web-based and mobile information systems.

III. Automated Hyponymy Extraction

We now introduce our research questions, followed by our research method based on content analysis and Tregex.

RQ1. What are the different ways to express hyponymy in privacy policies, and what categories emerge to characterize the linguistic mechanisms for expressing hyponymy?

RQ2. What are the Tregex patterns that can be used to automatically identify hypernymy and how accurate are these patterns?

RQ3. What percentage of information type coverage can be extracted by applying the hyponymy patterns to privacy policies?

RQ4. How does hypernymy vary across policies within a single domain, and across multiple domains?

Figure 1 presents an overview of our approach to answer the research questions. During Steps 1 and 2, the analyst prepares the input text to the NLP tools used in Steps 3 and 4, and to the crowdworker platform in Step 6, which is based on Amazon Mechanical Turk (AMT).

Steps 1-2 are performed manually by an analyst, once for each policy, which requires 30-90 minutes per policy. In Step 1, the input text begins as a text file, which can be extracted from HTML or a PDF document. In Step 2, the analyst itemizes the text into paragraphs consisting of 50-120 words, while ensuring that each paragraph's context remains undivided. This invariant can lead to paragraphs that exceed 120 words, which are balanced by smaller 50-60 word paragraphs. The 120-word limit is based on the average time required by one crowdworker to identify information types in Step 6, which averages 60 seconds [5].

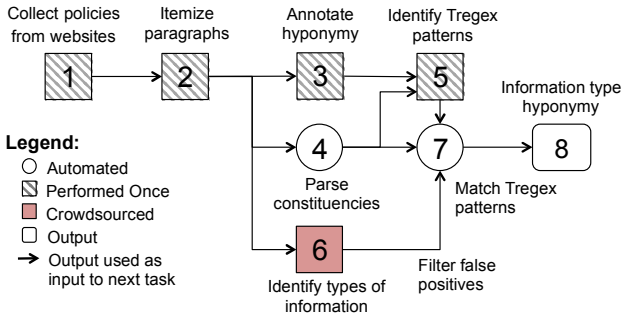

Fig. 1. Hyponymy Extraction Framework

In Step 3, two or more analysts perform content analysis on the same 120-word paragraphs from Step 2 to manually identify and categorize hyponymy in privacy policy text. The analysts meet periodically to agree on heuristics and guidelines for annotating hyponymy, before combining the hyponymy annotations with corresponding constituency parses from Step 4 to infer matching Tregex patterns in Step 5. The Tregex patterns are identified in a manual, interpretive process performed once: the patterns are then used in an automated Step 7 to find hyponymy relationships.

The Tregex patterns are generic and do produce false positives. To filter out false positives, we use crowdworker annotations produced in Step 6. If a Tregex pattern matches a phrase that has not been annotated by at least two crowdworkers as an information type, that match is discarded as a false positive. We describe each of these steps in detail in the following sub-sections.

A. Annotating Hyponymy

Research question RQ1 asks how hyponymy appears in privacy policies in the wild. To answer RQ1, we selected 30 privacy policies across six domains: shopping, telecom, social networking, employment, health, and news (see Table I). These policies are part of a US-centric convenience sample, although, we include a mix of shopping companies who maintain both online and brick-and-mortar stores, and we chose the top telecom websites and five top social networking websites in the US. Table I presents the 30 policies in our development set and 12 policies in our test set by category and date last updated.

TABLE I. PRIVACY POLICY DATASETS FOR HYPONYMY STUDY

	Company's Privacy Policy	Industry Category	Last Updated
Development Dataset	CareerBuilder	Employment	5/18/14
	Glassdoor	Employment	9/9/14
	Indeed	Employment	2015
	Monster	Employment	3/31/14
	SimplyHired	Employment	4/21/10
	23andme	Health	3/25/13
	HealthVault	Health	11/2013
	Mayo Clinic	Health	7/13/13
	MyFitnessPal	Health	6/11/13
	WebMD	Health	5/6/13
	ABC News	News	11/18/16
	Accuweather	News	10/17/13
	Bloomberg	News	7/15/14
	Reuters	News	11/2011
	WashPost	News	2010
	Barnes and Noble	Shopping	5/7/13
	Costco	Shopping	12/31/13
	Lowes	Shopping	4/25/15
	Over Stock	Shopping	1/9/13
	Walmart	Shopping	9/17/13
	Facebook	Social Networking	4/9/13
	Kik	Social Networking	9/22/14
	LinkedIn	Social Networking	10/23/14
	SnapChat	Social Networking	11/17/14
	Whatsapp	Social Networking	7/7/12
	AT&T	Telecom	9/16/13
	Charter Comm.	Telecom	5/4/09
	Comcast	Telecom	3/1/11
	Time Warner	Telecom	9/2012
	Verizon	Telecom	10/2014
Test Dataset	Dice	Employment	2/14/17
	USJobs	Employment	2015
	CVS	Health	1/13/16
	Fitbit	Health	12/9/14
	CNN	News	7/31/15
	Fox News	News	10/26/16
	JCPenny	Shopping	5/22/15
	Nordstrom	Shopping	10/1/15
	Twitter	Social Networking	5/18/15
	Whisper	Social Networking	5/22/15
	Sprint	Telecom	3/29/17
	T-mobile	Telecom	12/31/16

The policies are first prepared by removing section headers and boilerplate language that does not describe relevant data practices, before saving the prepared data to an input file for an AMT task, as described by Steps 1 and 2 in Figure 1. The task employs an annotation tool developed by Breaux and Schaub [5], which allows analysts to select relevant phrases matching a category. The analysts are asked to annotate three types of phrases for each hyponymy relationship identified: a *hypernym phrase*, which describes the general category phrase; one or more *hyponym phrases*, which describe members of the category; and any *keywords*,

which signal the hyponymy relationship. For example, in Figure 2, the phrase "personal information" is the hypernym, which is followed by the keywords "for example," which indicate the start of a clause that contains the hyponyms "name," "address" and "phone number."

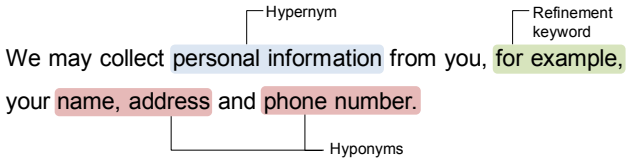

Fig. 2. Hyponymy Annotations

The annotation process employs two-cycle coding [27]. In the first cycle, the policies are annotated to identify the prospective hyponym patterns, after which the second-cycle is applied to group these patterns into emergent categories. The two-cycle coding process begins with an initial set of five policies, during which guidelines and examples are developed by the analysts to improve consistency and to support replication. Next, the analysts meet to discuss their initial results, to reconcile differences and to refine the guidelines. After agreeing on the guidelines and initial categories, the analysts annotate the remaining policies, before meeting again to reconcile disagreements and measure the kappa.

The itemized policy paragraphs are also used as input for crowdsourced annotations, as described in Step 6. The purpose of the crowd's annotations is to identify all relevant information types as "information." This task is similar to the analysts' annotation task; however, the annotation "information" is not as specific as the annotations for hypernym, hyponymy, and refinement keyword. The crowdworkers would annotate the sentence in Figure 2 with "personal information," "name," "address," and "phone number" as information types. Each paragraph that the analysts annotate is also annotated by five crowdworkers on AMT. If at least two of the crowdworkers annotate a phrase as "information," it is considered a valid annotation. Each valid annotation from the crowd can be used as a type of validation for matching Tregex patterns.

During annotation, the analyst may encounter nested hypernymy, which occurs when a hypernym-keyword-hyponym triple has a second triple embedded within the phrase, often within the hyponym phrase. For example, the sentence in Figure 3: "Self-reported information includes information you provide to us, including but not limited to, personal traits (e.g., eye color, height)" contains three nested hyponymy relations. The phrase "information you provide to us" is the hyponym of "self-reported information," and it is also the hypernym of "personal traits." Similarly, "personal traits" is the hyponym of "information you provide to us" and it is the hypernym of "eye color, height." To correctly extract the hyponym-hypernym pairs, we first coded the annotated phrases in numerical order, then we represented hypernym-keyword-hyponym triples using a three-character alphanumeric sequence that corresponds to the order of phrases in the sentence; repeated numbers represent the same phrase in one or more relations. For example, the sentence in Figure 3 would have the code "123; 345; 567" wherein the "3" represents "information you provide to us" as the hyponym in the first relation and the hypernym in the second relation, and "5" represents "personal traits."

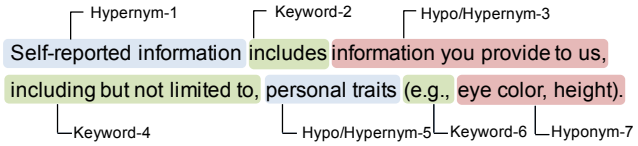

Fig. 3. Nested Hyponymy

B. Identifying Tregex Patterns

Tregex is a language for matching subtrees in a constituency parse tree [18]. The Tregex patterns are created by examining the parse tree for each annotated sentence in a hyponymy category. Figure 4 presents the constituency parse tree for the sentence from Figure 2. The colors in the figure show which part of the parse tree matches which part of the Tregex pattern. In the parse tree, the root tree node labeled ROOT appears in the upper, left-hand corner with a single, immediate child labeled S; each child is indented slightly to the right under the parent, and siblings are indented equidistant from the left-hand side of the figure. The matching Tregex pattern below the parse tree has three parts: a noun phrase (NP) that is assigned to a variable named "hypernym" via the equal sign (in blue), followed by a dollar sign that indicates a sibling pattern, which is the keyword phrase (in green), followed by a less-than sign that indicates an immediate child node, which is another NP assigned to the variable "hyponym" (in red). Tregex provides a means to answer RQ2 by expressing patterns that match the annotated hypernyms and hyponyms and their lexical coordination by the keywords.

We developed a method to write Tregex patterns to match hyponymy. Given a constituency parse tree, the first step is to traverse the tree upwards from each hypernym and hyponym until you find a shared ancestor node that bridges the two constituents. In Figure 4, the verb (VB) "collect" is an immediate child of the reference node verb phrase (VP). The VP is not present in the Tregex pattern, though it is the reason the NP and the prepositional phrase (PP) are defined as sister nodes ($) in the matching Tregex pattern also in Figure 4. The reference node is omitted in the Tregex pattern to keep the pattern in its most general form. Once it is established that we can relate two parts of the tree in one pattern, we traverse the two subtrees back down until we are able to isolate the constituents, in this case, a NP containing the hypernym and hyponym. When extracting the desired NP representing either the hypernym or hyponym we must maintain a level of generalizability. To do this we reference the NP using relationships between itself and ancestor nodes, such as a parenthetical phrase (PRN) as a sister of the NP it modifies or a NP as the immediate, right sister of the VB to

which it is the object. By encoding these relationships into the pattern versus directly copying the word order, we avoid over specification.

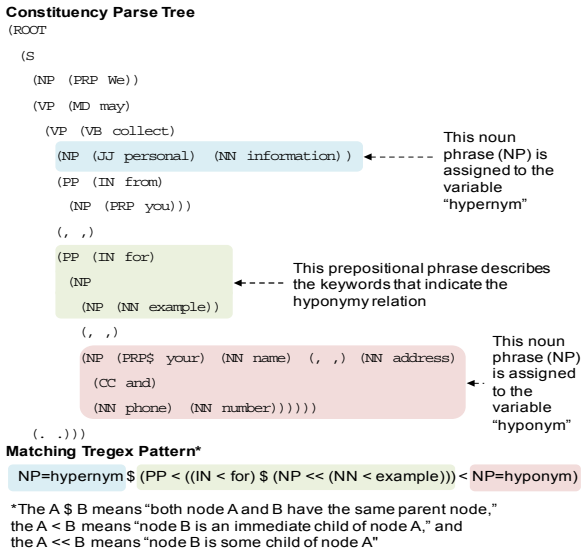

Fig. 4. Tregex Pattern Matcher

The development of Tregex patterns is a balance between generally characterizing the lexical relationships among words in a pattern, and specializing the pattern to avoid false positives. The pattern shown in Figure 4, which matches information type hyponyms after the "for example" keywords, can also match a data purpose hyponym: e.g., "we share your personal information for marketing, for example, product and service notifications." To automatically filter out such hyponyms that are not information type hyponyms, we use a lexicon constructed from crowd sourced information type tasks, described by Breaux and Schaub [5]. In the information type tasks, the crowdworkers are asked to annotate all information types in a given paragraph. For instance, in the privacy statement: "We collect your personal information such as your name, and address…", the crowdworkers would have annotated the phrases "personal information", "name" and "address" as information types.

We evaluated the Tregex patterns by comparing the number of hypernym-hyponym pairs identified across all 30 policies and comparing that to the pairs identified by trained analysts. While a pattern may produce a false-negative in one policy, it could find that pair in another policy. This evaluation strategy prioritizes our goal to extract a general ontology from multiple policies, over a separate goal to attribute hypernym-hyponym pairs to specific policies.

In addition, we developed a test dataset (see Table I) which we used to test the accuracy of our approach. The test set consists of 12 policies that were annotated by trained analysts. The test dataset was not used during the development of the Tregex patterns, and were annotated after we produced the Tregex patterns from the development dataset.

C. Ontological Completeness

The question RQ3 asks what percentage of information type coverage can be extracted by applying the Tregex patterns to privacy statements. To answer RQ3, we developed three types of lexicons – crowdworker lexicon, analyst lexicon and Tregex lexicon. The *crowdworker lexicon* consists of all the information types in our dataset of 30 privacy policies (see Table I). For the construction of this lexicon, we use the entity extractor developed by Bhatia and Breaux [1], which takes as input the crowd sourced tasks for each policy, where the crowdworkers have annotated all the information types in the policies [5]. The *analyst lexicon* is constructed by using the entity extractor on the analysts' hyponymy annotations described in Section III.A. The *Tregex lexicon* is constructed using the entity extractor on the information type hyponymy identified by the Tregex patterns, as described in Section III.B. We compare the *crowdworker lexicon, the analyst lexicon* and the *Tregex lexicon* to understand what percentage of information types are covered by the hyponymy patterns.

IV. Evaluation and Results

We now describe our results from the content analysis, Tregex pattern development and lexicon comparisons.

A. Hyponymy Taxonomy from Content Analysis

The first and second authors annotated the 30 policy development dataset shown in Table I. This process consumed 10 and 11 hours for each annotator, respectively, and yielded 304 annotated instances of hyponymy after final reconciliation. The guidelines that were developed can be summarized as follows: only annotate information type noun phrases; annotations should not span more than a single sentence and should include any modifying prepositional or verb phrases that qualify the information type; and if the noun phrase is an enumeration, annotate all noun phrases together.

The second-cycle coding to categorize the hyponymy relationships was based first on the refinement keyword semantics, and next based on the relative order of the hypernym (H), keyword (K) and hyponym (O) in the privacy policy text. We answer RQ1 by defining the following resulting categories:

- *Incomplete Refinement (Inc.)*: The keywords suggest that the hyponymy consists of an incomplete subset of the phrases that can be used as hyponyms for the given hypernym. For instance, the keywords "such as" and "including" indicate that the given hyponyms are part of an incomplete list.
- *Complete Refinement (Com.)*: The keywords indicate that the hyponyms are the complete list that belong to the hypernym. For instance, the keywords, "consists of" and "i.e." indicate that the given list of hyponyms are complete for the respective hypernym.

- *Implied Refinement (Imp.)*: The refinement keyword is a punctuation such as a colon (:) or dash (-) and indicates that there is an implied hyponymy.

The resulting syntactic categories are defined as follows:

- *HKO* – The hypernym occurs first, followed by the keyword, followed by the hyponym. This pattern is predominantly used to illustrate examples (hyponyms) of leading technical words (the hypernym).
- *OKH* – The hyponym occurs first, followed by the keyword, followed by the hypernym. This category describes lists in which the last term generalizes the preceding terms.
- *HO* – The hypernym occurs first followed by the hyponym, and there is no keyword. This category is found when the hypernym is the section header, followed by a subsection of implied hyponyms; there are no keywords that explicitly indicate the hyponymy.
- *KHO* – The keyword occurs first, followed by the hypernym, followed by the hyponym. This category is rare and uses a colon to separate the hypernym from a list of hyponyms.

We measured the degree of agreement above chance using Fleiss' Kappa [10] for the hyponym categories from the second-cycle coding. Each hyponymy instance is assigned a semantic category and a syntactic category. The Kappa was computed using the composition of categories. For example, a hyponymy relationship that belongs to the incomplete semantic category and HKO syntactic category is assigned to the category combination of {Inc.-HKO}. The Fleiss Kappa for all mappings from annotations to hyponym categories and the two analysts was 0.99, which is a very high probability of agreement above chance alone.

Table II and III presents the keyword taxonomies for the semantic and syntactic categories, respectively: including the *Category*, the *Refinement Keywords* that help detect the hyponymy, and the proportion of annotations in the category across all 30 policies (*Freq.*). The most frequent category among the semantic categories was *incomplete refinement*.

TABLE II. KEYWORD TAXONOMY FOR SEMANTIC CATEGORIES

Category	Refinement Keywords	Freq.
Incomplete Refinement	such as, such, include, including, includes, for example, e.g., like, contain, (and|or|any | as well as any| certain) other, concerning, relating to,, is known as, classifies as	96.05%
Complete Refinement	consists of, is, i.e., either, constitute, of your, following types of, in	2.96%
Implied Refinement	(), :, -, . (section header)	0.98%

TABLE III. KEYWORD TAXONOMY FOR SYNTACTIC CATEGORIES

Category	Refinement Keywords	Freq.
HKO	such as, such, including, for example, include, includes, concerning, is, e.g., like, i.e., of your, contain, relating to, that relates to, generally not including, consists of, concerning, either, ( ), :, -	88.48%
OKH	(and|or|any| as well as any| certain) other, constitute, as, other, is known as, classifies as, is considered	10.52%
HO	None	0.66%
KHO	following types of	0.33%

H: Hypernym, O: Hyponym, K: Keyword

TABLE IV. FREQUENCY OF HYPONYMY CATEGORIES

Syntactic Categories	Semantic Categories			
	Inc.	Com.	Imp.	Total
HKO	261	7	1	269
OKH	30	2	0	32
HO	0	0	2	2
KHO	1	0	0	1
Total	292	9	3	304

H: Hypernym, O: Hyponym, K: Keyword; Inc.: Incomplete Refinement, Com.: Complete Refinement, Imp.: Implied Refinement

B. Tregex Pattern Evaluation

We identified a total of 72 Tregex patterns to answer RQ2, which can be used to automatically identify hyponymy in privacy policies. Due to space limitations, we only present an example subset of Tregex patterns in Table V. The HO syntactic category, which has no keywords, cannot be reliably characterized by a high precision pattern, i.e., low false positives.

TABLE V. TREGEX PATTERNS

Hyponym Category	Tregex Pattern Example	# Tregex Patterns
HKO	NP=hypernym $ (VP < (VBZ < includes) < NP=hyponym)	60
OKH	NP=rhs < (CC < or|and) < (NP=lhs < (JJ < other))	11
KHO	("NP < (NP < ((NP < (JJ < following) < (NNS < types)) $. (PP < (IN < of) < NP=hypernym)) $.. NP=hyponym)	1

We evaluate our Tregex patterns as described in Section III.B. Table VI presents the evaluation precision and recall in terms of the automated hyponym extraction results compared to the analyst annotations for the 30 policies. The identified instance of hyponymy is counted as a true positive (TP), only if the hypernym and the hyponym both match the analyst annotations. Otherwise, it is counted as a false positive (FP). For example, a successful extraction applied to the phrase "online activity e.g., sites visited, pages visited," yields two TPs: 1) "sites visited" is a kind of "online activity," and 2) "pages visited" is a kind of "online activity." The results in Table VI were computed using the crowdworkers' information type annotations as a means to filter out FPs as described in Section III.B; without this filtering, the average precision drops from 0.72 to 0.22.

TABLE VI. EVALUATIONS OF TREGEX PATTERNS

	Privacy Policy	Precision	Recall
Development Dataset	CareerBuilder	0.84	0.78
	Glassdoor	0.76	0.56
	Indeed	0.70	0.60
	Monster	0.87	0.93
	SimplyHired	0.51	0.80
	23andme	0.57	0.69
	HealthVault	0.97	0.97
	Mayo Clinic	0.72	0.53
	MyFitnessPal	0.35	0.70
	WebMD	0.86	0.86
	ABC News	0.62	0.62
	Accuweather	0.65	0.69
	Bloomberg	0.90	0.80
	Reuters	1.00	0.92
	WashPost	0.80	0.89
	Barnes and Noble	0.65	0.53
	Costco	0.85	0.92
	Lowes	0.50	0.61
	Over Stock	0.83	0.56
	Walmart	0.73	0.91
	Facebook	0.49	0.53
	Kik	0.66	0.79
	LinkedIn	0.65	0.59
	SnapChat	0.61	0.53
	Whatsapp	0.92	0.96
	AT&T	0.67	0.72
	Charter Comm.	0.78	0.78
	Comcast	0.76	0.76
	Time Warner	0.90	0.90
	Verizon	0.55	0.74
	Average	**0.72**	**0.74**
Test Dataset	Dice	0.79	0.77
	USJobs	0.63	0.83
	CVS	0.33	0.50
	Fitbit	0.30	0.29
	CNN	0.40	0.49
	FoxNews	0.71	0.71
	JCPenny	0.64	0.47
	Nordstrom	0.29	0.48
	Twitter	0.58	0.58
	Whisper	0.75	0.33
	Sprint	0.32	0.60
	T-mobile	0.34	0.55
	Average	**0.51**	**0.55**

To answer RQ3, we compiled an ontology of hypernym-hyponymy pairs identified by the Tregex patterns and compared these to the pairs found by the analysts across all 30 policies in the development dataset. The average precision for this evaluation is 0.72, and the average recall is 0.74. The same patterns were matched to the test dataset, which was annotated by the third and fourth authors. The average precision for this evaluation is 0.51 and the average recall is 0.55. The difference in results between the development and test dataset is primarily due to slight variations in the tree structure which deviate from existing patterns. One example of a slight variation in structure is the sentence: "We may collect information about your device such as the type, operating system details, signal strength, whether it is on and how it is functioning, as well as information about how you use the device and services available through it, such as your call and data usage and history, your location, web sites you have visited, applications purchased, applications downloaded or used, and other similar information." In this example, there are two instances of hyponymy relations indicated by the "such as" keywords (underlined). This sentence produces multiple FPs due to the fact that the first relation's hyponym phrase, "type, operating system details, signal strength…" also contains the second relation's hypernym: "…as well as information about how you use the device," where "information about how you use the device" is the hypernym in the second relation. The specialized patterns needed to extract hyponyms in nested phrases are not included among our Tregex patterns. Another explanation for lower precision and recall in the test dataset is new keywords were found, including the keyword "means."

We analyzed the FPs and false negatives (FNs) produced by the Tregex patterns to explain the low recall. One reason for the inaccurate identification of either the hypernymy or hyponymy is the misconstructed parse tree generated by the Stanford Parser. This can be due to the presence of syntactic ambiguity, where the modifier phrase can be attached to any of the preceding noun phrases. For example, the statement, "So for those we develop a more precise estimate of location by associating the serving cell tower ID with other information, like the latitude and longitude of the tower, radio frequency parameters, GPS information and timing differences in radio signals," the Stanford Parser attaches the noun phrase, "precise estimate of location" to the modifier phrase "like the latitude…" and, therefore, a Tregex pattern identifies this noun phrase to be the hypernym of the modifier. In contrast, the analysts annotated this phrase as the hyponym of "other information." The analysts were able to disambiguate the attachment based on the context and their domain knowledge. Another explanation for incorrect or missed identification of hyponyms is an incorrect POS-tag produced by the parser. It is reported that the sentence accuracy of the Stanford Parser is 56% [20]. Terms such as "zip" in "zip code" and "email" are frequently tagged as verbs. This creates the presence of a verb phrase constituent rather than a noun phrase, ultimately prohibiting noun phrase extraction. Certain health policies contain complicated phrases, e.g. numerous parenthetical examples, which are too complex to parse correctly, lowering precision and recall. Social networking policies utilize colloquial diction that was also complex to parse.

Our true positives also include incomplete identification of the hypernyms due to the presence of anaphora pronouns. For example, the sentences "We collect your personal information. *This* includes your name, address…" contains the pronoun "this," which refers to the noun phrase, "personal information" in the previous sentence. Our automated approach is limited to the sentence-level.

C. Comparison to Lexicon

As described in Section III.C, we constructed our *crowdworker lexicon* using the entity extractor [1] on the crowd-sourced tasks [5] for the 30 policies in our dataset (see Table I). This dataset has a total of 1,905 unique phrases. The *analyst lexicon*, which was constructed by using the entity extractor on the annotator hyponymy annotations, contains 677 phrases. The *crowdworker lexicon* contains 534 out of these 677 phrases, which converts to 28% of the total information types identified by the crowdworkers in the privacy policies using the analyst annotations. The difference in 143 phrases are false negatives (FN) that were identified by the analysts during hyponymy annotations, but were missed by the crowdworkers during the information type annotation tasks. The FN information types contain some information types that have uncommon meaning, for example "public profile" and were difficult to identify. The FNs also contain information types that are different forms of the information types already existing in the crowdworker lexicon, for instance, "new personal information" and "similar account information."

The *Tregex lexicon*, constructed by applying the entity extractor to the hyponymy identified by the Tregex patterns yielded 614 phrases. The *Tregex lexicon* shared 458 phrases with the *crowdworker lexicon*, which yields 24% of the information types identified by crowdworkers in the privacy policies. The difference of 156 phrases found by the Tregex patterns were false positives (FP), in addition to 60 phrases that were included as true positives (TPs), and were instead missed by the crowdworkers. For example, the phrase, "aggregate demographic information" is present in the *Tregex lexicon*, but is missing from the *crowdworker lexicon*.

On comparing the *Tregex lexicon* with the *analyst lexicon*, we found that 528 phrases were shared between both the lexicons.

D. Ecological Assessment of Type Semantics

We now report findings from analyzing the hypernyms across policies and across human subject interpretations. The complete dataset is available online.[1]

1. Semantic Variation across Domain Policies

Among the 30 policies studied, few policies shared hypernym terms: while 27 policies shared the hypernym information, an average 1.56 policies shared terms. In several instances, the hypernyms across two or more policies were semantically related: "contact information" and "contact data" are close synonyms, whereas "equipment identifiers" are subordinate to "equipment information." The following categories were shared by five or more policies: demographic information, personal information, contact information, payment and billing information, and device and equipment information.

[1] http://gaius.isri.cmu.edu/dataset/hyper17/

Figure 5 shows different interpretations of demographic information across six policies: two employment companies (Career Builder, Simply Hired), two telecommunications companies (AT&T, Verizon), and two health companies (MyFitnessPal, Mayo Clinic). Mayo Clinic has a different, outstanding interpretation of the hypernym to include name, address, and telephone number, which appear as "personal information" in Career Builder's policy and as "billing information" in AT&T's policy. The remaining five interpretations in Figure 5 share the concept "gender" and include variations on Zip or postal code and birthday or age, which are used to target individuals with advertising. MyFitnessPal includes "birthday," which is the most privacy sensitive, whereas Career Builder, Simply Hired and AT&T include "age," which is less sensitive, and Verizon includes "age range," which is least sensitive.

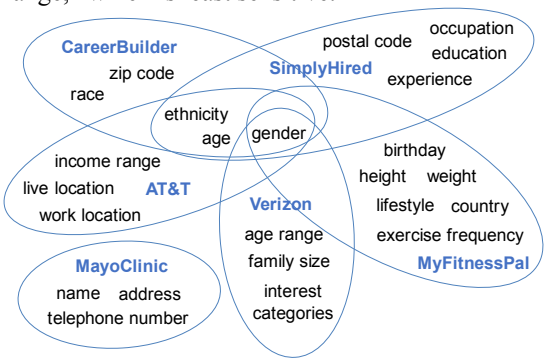

Fig. 5. Venn diagram illustrating shared hyponyms for the hypernym demographic information

In addition to using information type generalization to reduce sensitivity, demographic category members may be related by how the company uses the information. In Figure 5, AT&T has "income range" and Simply Hired has "occupation," which can be used to estimate income range. Verizon includes "interest categories," and MyFitnessPal includes "lifestyle," which are both attributes used in advertising platforms, such as Nielsen MyBestSegments®. Other hypernyms, such as contact and payment information, also appear to correspond to the purposes for which data will be used (e.g., to contact individuals, or pay account balances).

V. Discussion and Future Work

We now discuss our results and their impact on improving understanding of privacy practices and on extracting goals from privacy policies. Hyponymy is one among other relationships, such as meronymy and synonymy, that is used to describe the relationships between phrases in an ontology. Using our approach, we were able to identify hypernym-hyponym pairs with an average precision of 0.72 and average recall of 0.74, as compared to the pairs identified by the analysts. We evaluated our approach using a test dataset of 12 policies, which resulted in an average precision of 0.51 and an average recall of 0.55. We believe this is the first step

towards automating the construction of an information type ontology for privacy policies, by reducing the search space of possible relationships that exist between two phrases in a given lexicon. For the 1,905 information types identified by crowdworkers in the 30 policy development dataset, there are 1.81M such possible comparisons needed to construct a complete ontology. In addition, to determine the relationship between the 458 shared terms between the Tregex lexicon and the crowdworker lexicon, we would have needed 104,653 paired comparisons, which we were able to identify automatically.

Our approach has implications beyond the current dataset and examples. First, we believe the approach can be applied to other legal documents and user scenarios, in which technical terminology should be elaborated by example. However, our approach would be less effective for the reader in domains in which the readers share the same tacit knowledge of technical terminology, and thus they would be less informed by hyponymy. Second, despite the observed tool limitations, we believe that continued development of the existing Tregex patterns could lead to a gold standard for patterns in information type hyponymy extraction. The automatically extracted hypernymy from privacy policies using our approach, while limited in regard to precision and recall, could be manually inspected to remove false positives and add missing false negatives, to further build a large corpus of information type hypernyms that could be used as training data for advanced machine learning. Finally, our analysis of the variation among hyponyms suggests that the hypernym patterns could be used to develop controlled natural language templates for policy authors to write hypernymy in privacy policies, which could be easily processed automatically. Alternatively, creating a dictionary of information type hypernyms that could accompany privacy policies as appendices would allow for strict regulation of information type terms to help maintain consistency within and across industries.

Our results also show a limitation in existing NLP-based tools that rely on established lexical databases. We observed that the vocabulary used in the 30 privacy policies is very different from the vocabulary used in the popular lexical database WordNet: about 17% of the information types that we found in the 30 policies were present in the WordNet, meaning 83% of the information types are more precise terms or missing from WordNet, altogether. This finding suggests that existing NLP-tools that rely on WordNet could be adapted to privacy with a new lexical database for privacy. As future work, we plan to integrate our results into such a database useful for privacy requirements analysis.

As future work, we also plan to use free-listing to investigate how system users and data subjects vary in their interpretation of hypernyms within and across domains. Free-listing is a technique from psychology in the 1950s used to examine associations among concepts [3]. Free-listing allows for responses that are uninfluenced by the prompts that elicit them and promote a maximum number of responses without eliciting random responses. Brewer describes three techniques for free-listing: *nonspecific prompting*, which asks participants to state examples of a kind; *reading back,* which lists the previous responses back to the participant to check whether a participant can see any omissions in their original list; and *semantic cueing*, which asks participants to identify concept names that are related to a given name [7]. Brewer et al. found that follow-up probes increased responses by 7% [8], and semantic cues increased the number of items by 48-49% [9]. Using the three free-listing techniques, we plan to elicit a broad range of domain terminology at multiple levels of hyponymy.

In addition, we plan to use other forms of natural language processing to address limitations in the current approach. This includes using the dependency parser as an alternative to constituency parser to identify hyponymy relations among phrases. A dependency parser would allow for a deeper understanding of the relationships between individual words, specifically, how each word depends on other words in the sentence to convey meaning and formulate structure. We believe that the type of dependencies between hypernym and hyponym phrases may provide insight into patterns of higher detail than Tregex patterns. Furthermore, we envision using our empirically validated hyponymy annotations as a training set for machine learning algorithms to develop domain specific models of hyponymy. Co-reference resolution algorithms can be used to increase the accuracy of the extracted hypernymy relationships for statements that have anaphora.

Finally, we plan to integrate this result into emerging formal methods that depend on precise descriptions of information types and their semantic relatedness. For example, Slavin et al. have used a privacy ontology to link privacy policy statements to mobile application API calls [28] and Breaux et al. have shown how privacy ontology can be used with Description Logic to check a company's data practices for compliance with the OECD collection and use limitation principles [6].

ACKNOWLEDGMENT

We thank the CMU RE Lab for their helpful feedback. A short version of this paper based on a smaller dataset of 15 policies appears in the proceedings of the IEEE Workshop on Artificial Intelligence and Requirements Engineering [2]. This research was funded by NSF Frontier Award #1330596, NSF CAREER Award #1453139 and NSA Award #141333.

REFERENCES

[1] J. Bhatia, T.D. Breaux, "Towards an information type lexicon for privacy policies." *IEEE 8th Int'l W'shp Req'ts Engr. & Law*, pp. 19-24, 2015.

[2] J. Bhatia, M.C. Evans, S. Wadkar, T.D. Breaux, "Automated Extraction of Regulated Information Types using Hyponymy Relations," *IEEE 3rd International Workshop on Artificial Intelligence for Requirements Engineering (AIRE),* pp. 19-23, Aug. 2016.

[3] W. Bousfield, W. Barclay, "The relationship between order and frequency of occurrence of restricted associative responses," *Journal of Experimental Psychology*, 40, pp. 643–647, 1950.

[4] T.D. Breaux, H. Hibshi, A. Rao. "Eddy, a formal language for specifying and analyzing data flow specifications for conflicting privacy requirements." *Req'ts Engr. J.*, 19(3): 281-307, 2014.

[5] T.D. Breaux, F.Schaub, "Scaling requirements extraction to the crowd: experiments on privacy policies." *IEEE 22nd Int'l Req'ts Engr. Conf.*, pp. 163-172, 2014.

[6] T.D. Breaux, D. Smullen, H. Hibshi. "Detecting repurposing and over-collection in multi-party privacy requirements specifications." *IEEE 23rd Int'l Req'ts Engr. Conf.,* pp. 166-175, 2015.

[7] Brewer, D. D. (2002). Supplementary interviewing techniques to maximize output in free listing tasks. *Field methods*, *14*(1), 108-118.

[8] D.D. Brewer, S.B. Garrett, S. Kulasingam. "Forgetting as a cause of incomplete reporting of sexual and drug injection partners." *Sexually Transmitted Diseases* 26:166–76, 1999.

[9] D.D., Brewer, S.B. Garrett, G. Rinaldi. "Free listed items are effective cues for eliciting additional items in semantic domains." *Applied Cognitive Psych.*, 16(3): 343–358, 2002.

[10] J.L. Fleiss, "Measuring nominal scale agreement among many raters," *Psychological Bulletin*, n. 76, pp. 378-382.

[11] S. Ghanavati, *Legal-URN framework for legal compliance of business processes.* Ph.D. Thesis, University of Ottawa, 2013.

[12] M. A. Hearst, "Automatic acquisition of hyponyms from large text corpora," in proceedings of the 14th conference on Computational linguistics - Volume 2 (COLING '92), Vol. 2. Association for Computational Linguistics, 1992, pp. 539-545. DOI=http://dx.doi.org/10.3115/992133.992154

[13] M. Hellman, N.A. Smith, "Good question! statistical ranking for question generation," *Annual Conf. North Amer. Chapter of the. ACL*, pp. 609–617, 2010.

[14] M. Jackson, P. Zave, "Domain descriptions," *IEEE Symp. Req'ts Engr*, pp. 56-64, 1993.

[15] D. Jurafsky, J.H. Martin, *Speech and language processing: An introduction to natural language processing, computational linguistics and speech recognition*. Prentice Hall, 2009.

[16] G. Koliadis, N.V. Desai, N.C. Narendra, A.K. Ghose. "Analyst-mediated contextualization of regulatory policies," *IEEE Int'l Conf. Services Comp.,* pp. 281-288, 2010.

[17] A. van Lamsweerde, "Formal specification: a roadmap," Int'l Soft. Engr. Conf., pp. 147-159, 2000.

[18] R. Levy, G. Andrew. "Tregex and Tsurgeon: tools for querying and manipulating tree data structures." *5th Int'l Conf. on Lang. Res. & Eval.* (LREC 2006).

[19] W. Li, J. Huffman Hayes, M. Truszczynski, "Torwards more efficient requirements formalization," REFSQ, *LNCS* v. 9013, pp. 181-197, 2015.

[20] Manning, Christopher. "Part-of-speech tagging from 97% to 100%: is it time for some linguistics?" *Computational Linguistics and Intelligent Text Processing* (2011): 171-189.

[21] Manning, Christopher D., Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 55-60.

[22] J. Maxwell, A. Antón. "Developing production rule models to aid in acquiring requirements from legal texts," *17th IEEE Int'l Req'ts Engr. Conf.*, pp. 101-110, 2009.

[23] G. A. Miller. "WordNet: A Lexical Database for English." *Comm. ACM* 38(11): 39-41, 1995.

[24] E. Paja, F. Dalpiaz, P. Giorgini. "Modelling and reasoning about security requirements in socio-technical systems," *Data & Know. Engr.* v. 98, pp. 123-143, 2015.

[25] E. Rosch, "Principles of categorization," Rosch & Lloyd (eds), *Cognition and Categorization*, pp. 27-48, 1978.

[26] K. Ryan, "The role of natural language in requirements engineering," *IEEE Symp. Req'ts Engr*, pp. 240-242, 1993.

[27] J. Saldaña. *The Coding Manual for Qualitative Researchers*, SAGE Publications, 2012.

[28] R. Slavin, X. Wang, M.B. Hosseini, W. Hester, R. Krishnan, J. Bhatia, T.D. Breaux, J. Niu. "Toward a Framework for Detecting Privacy Policy Violation in Android Application Code," *ACM/IEEE 38th International Conference on Software Engineering*, pp. 25-36, 2016.

[29] R. Snow, D. Jurafsky, and A. Y. Ng, "Learning syntactic patterns for automatic hypernym discovery," in Advances in Neural Information Processing Systems, 2005.

[30] S. Tratz, E.H. Hovy. "Summarization Evaluation Using Transformed Basic Elements." *Text Analytics Conference (TAC-08),* NIST, 2008.

[31] A. Tversky. "Features of similarity." *Psych. Review*, 84(4): 327-352, 1977.