Enhancing Readability of Privacy Policies Through Ontologies

by

Dhiren Audich

A Thesis
Presented to
The University of Guelph

In partial fulfillment of requirements
for the degree of
Master of Science
in
Computer Science

Guelph, Ontario, Canada

© Dhiren Audich, February, 2018
ABSTRACT

ENHANCING READABILITY OF PRIVACY POLICIES THROUGH ONTOLOGIES

Dhiren Audich
University of Guelph, 2018

Advisors:
Dr. Blair Nonnecke
Dr. Rozita Dara

Privacy policies operate as memorandums of understanding (MOUs) between the users and providers of online services. Research suggests that users are deterred from reading policies because of their length, difficult language, and insufficient information. Users are more likely to read short excerpts if they immediately addresses their concerns.

As a first step in helping users find pertinent information in privacy policies, this thesis presents the development of a domain ontology using natural language processing (NLP) algorithms as a way to reduce costs and speed up development.

By using the ontology to locate key parts of privacy policies, average reading times were substantially reduced from 8-12 minutes to 45 seconds. In the process of extracting keywords from the privacy policy corpus, a supervised NLP algorithm performed marginally better (7%) but showed greater promise with larger training sets. Additionally, trained non-domain experts achieved a combined F1-score of 71% when compared to a domain expert, and did so when extracting keywords from fewer policies.
Dedicated to Amma, Dadima, and Chitra-masi.
Completing my Masters was a grand undertaking and it wouldn’t have been possible without the grace and generosity of the following key individuals.

First and foremost, I would like to thank my parents, for sowing the seeds of inquisitiveness and inquiry in me which led me on a path to becoming a scientist. Your support and encouragement mean the world to me. Next, I would like to extend my heartfelt gratitude to my co-advisors: Dr. Rozita Dara and Dr. Blair Nonnecke, who have, for one reason or another, permitted me an almost abusive amount of time and flexibility in completing my Thesis. Were it not for their patience and guidance, I would have found it very difficult to complete my graduate studies. Individually, I would like to thank Rozita for her vast knowledge that helped shape my research, and Blair for his invaluable insights and long discussions about design and HCI.

Additionally, I would like to thank the following set of individuals who either collaborated with me or provided their valuable input: Dr. Luiza Antoine for reviewing my Thesis and providing feedback; Dr. Deborah Stacey for mentoring me with ontologies and reviewing my Thesis; Niharika Guntamukkala for providing the initial data set of privacy policies; Curtis Rasmussen for providing me with the data set with the user concerns; and Nikhil Sapru for helping me recruit participants for my user testing.

Lastly, I would like to thank everyone else who have influenced me over the course of my graduate studies; you know who you are, I thank you for shaping me in one way or another and supplying me with the courage to carry my own convictions.
# Table of Contents

List of Tables \hspace{1.2cm} viii

List of Figures \hspace{1.2cm} ix

List of Listings \hspace{1.2cm} xi

List of Abbreviations \hspace{1.2cm} xii

1 Introduction \hspace{1.2cm} 1
   1.1 Privacy Policies \hspace{1.2cm} 2
   1.2 Ontologies \hspace{1.2cm} 5
      1.2.1 What is an Ontology? \hspace{1.2cm} 5
      1.2.2 Building Ontologies \hspace{1.2cm} 6
      1.2.3 Evaluating Ontologies \hspace{1.2cm} 7
   1.3 Taxonomy Generation \hspace{1.2cm} 8
      1.3.1 Keyword & Keyphrase Extraction \hspace{1.2cm} 8
      1.3.2 Approaches to Keywords Extraction \hspace{1.2cm} 9
      1.3.3 Text Pre-Processing \hspace{1.2cm} 11
   1.4 Methodology \hspace{1.2cm} 12
      1.4.1 Problem Statement \hspace{1.2cm} 13
      1.4.2 Proposed Solution \hspace{1.2cm} 13
      1.4.3 Validation \hspace{1.2cm} 13
      1.4.4 Contribution \hspace{1.2cm} 15
      1.4.5 Thesis Organization \hspace{1.2cm} 16

2 Extracting Keywords From Online Privacy Policies With Unsupervised Learning Algorithms \hspace{1.2cm} 17
   2.1 Experiment I: Initial Benchmarking \hspace{1.2cm} 19
      2.1.1 Five Key Terms Extraction Methods \hspace{1.2cm} 20
      2.1.2 Experimental Setup \hspace{1.2cm} 23
      2.1.3 Results \hspace{1.2cm} 24
   2.2 Experiment II: Enlarged Dataset \hspace{1.2cm} 25
      2.2.1 Corpus \hspace{1.2cm} 25
      2.2.2 Results \hspace{1.2cm} 27
   2.3 Discussion \hspace{1.2cm} 27
# Extracting Keywords From Online Privacy Policies With Supervised Learning Algorithms

## Study Design

### Participants

- **Part I:**
  - **Procedure**
  - **Results**

### Experiment II: Supervised Learning

#### Part I

- **Part II**

## Discussion

## Conclusion

## A Domain Ontology For Online Privacy: DOOP

### Ontology Engineering

- **Methontology**
- **On-To-Knowledge**
- **Ontology 101**
- **RapidOWL**
- **NeOn**

### Methodology

#### Overview

- **Purpose**
- **Intended Use Cases**
- **Ontology requirements**
- **Construction**
- **Source for DOOP Development**
- **Competency questions**
- **Ontology specific classes**

### Results

#### Top structure

#### DOOP defining classes

#### Inference and structure

#### General class axioms

#### Validation

### Discussion

#### Generalizing keyword searches

# Conclusion

## Limitations & Future Work

## Privacy Information Framework

# References
A  Additional Results DOOP Validation  

B  Published paper  

B.1  Paper Information  

## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Comparing the results of stemming using Porter &amp; Lovin algorithms.</td>
<td>12</td>
</tr>
<tr>
<td>2.1</td>
<td>Breakdown of the 21 privacy policy corpus for Experiment I.</td>
<td>20</td>
</tr>
<tr>
<td>2.2</td>
<td>Criteria for manually extracting key terms.</td>
<td>21</td>
</tr>
<tr>
<td>2.3</td>
<td>Tuning TF-IDF with the document frequency (df) parameter.</td>
<td>24</td>
</tr>
<tr>
<td>2.4</td>
<td>Results for Experiment I: Applying keyword extraction algorithms to a small corpus of online privacy policies</td>
<td>25</td>
</tr>
<tr>
<td>2.5</td>
<td>Breakdown of the corpus of privacy policies for Experiment II.</td>
<td>25</td>
</tr>
<tr>
<td>2.6</td>
<td>Results for Experiment II: Applying keyword extraction algorithms to a corpus of online privacy policies</td>
<td>27</td>
</tr>
<tr>
<td>3.1</td>
<td>Results from manual keyword extraction by participants.</td>
<td>35</td>
</tr>
<tr>
<td>3.2</td>
<td>Comparing performance of manual extraction primary researcher vs. combined data set generated by participants</td>
<td>35</td>
</tr>
<tr>
<td>3.3</td>
<td>Comparing $F_1$-scores between participants’ and researcher’s data sets</td>
<td>35</td>
</tr>
<tr>
<td>3.4</td>
<td>Comparing performance of unsupervised vs. supervised learning algorithms for keyword extraction</td>
<td>36</td>
</tr>
<tr>
<td>3.5</td>
<td>Comparing $F_1$-scores between participants’ and researcher’s generated data sets from KEA</td>
<td>37</td>
</tr>
<tr>
<td>3.6</td>
<td>Comparing performance of keyword extraction from KEA based on primary researcher vs. combined data set generated by participants</td>
<td>37</td>
</tr>
<tr>
<td>4.1</td>
<td>17 concerns that users (children, teenagers, adults, seniors) most expressed about their online privacy</td>
<td>49</td>
</tr>
<tr>
<td>4.2</td>
<td>Breakdown of DOOP</td>
<td>51</td>
</tr>
<tr>
<td>4.3</td>
<td>Example of a class in DOOP</td>
<td>52</td>
</tr>
<tr>
<td>4.4</td>
<td>Object properties presently available in DOOP</td>
<td>54</td>
</tr>
<tr>
<td>4.5</td>
<td>Results for DOOP validation, experiment 1.</td>
<td>57</td>
</tr>
<tr>
<td>4.6</td>
<td>Results for DOOP validation, experiment 2.</td>
<td>57</td>
</tr>
<tr>
<td>4.7</td>
<td>Results for DOOP validation, experiment 3.</td>
<td>58</td>
</tr>
<tr>
<td>4.8</td>
<td>Results for DOOP validation, experiment 4.</td>
<td>58</td>
</tr>
<tr>
<td>4.9</td>
<td>Results for DOOP validation for query 2, experiment 1</td>
<td>65</td>
</tr>
<tr>
<td>4.10</td>
<td>Results for DOOP validation query 2, experiment 2</td>
<td>66</td>
</tr>
<tr>
<td>4.11</td>
<td>Results for DOOP validation for query 2, experiment 3</td>
<td>67</td>
</tr>
<tr>
<td>4.12</td>
<td>Results for DOOP validation for query 2, experiment 4</td>
<td>67</td>
</tr>
</tbody>
</table>
List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Overview of building an ontology for online privacy.</td>
<td>15</td>
</tr>
<tr>
<td>2.1</td>
<td>Experiment I: Running keyword extraction algorithms over a small corpus of privacy policies</td>
<td>19</td>
</tr>
<tr>
<td>2.2</td>
<td>Creating a baseline for the evaluation of keyword extraction algorithms.</td>
<td>20</td>
</tr>
<tr>
<td>2.3</td>
<td>Experiment II: Running keyword extraction algorithms over a corpus of online privacy policies</td>
<td>26</td>
</tr>
<tr>
<td>2.4</td>
<td>Sample of the output from TF-IDF</td>
<td>28</td>
</tr>
<tr>
<td>3.1</td>
<td>Experiment I: Training participants for extracting keywords from online privacy policies</td>
<td>33</td>
</tr>
<tr>
<td>3.2</td>
<td>Experiment II Part I: Testing KEA against manual extraction and AlchemyAPI</td>
<td>36</td>
</tr>
<tr>
<td>3.3</td>
<td>Experiment II Part II: Evaluating KEA results trained on non-domain experts’ data set of manually extracted keywords from privacy policies</td>
<td>38</td>
</tr>
<tr>
<td>4.1</td>
<td>Overview of how DOOP was constructed and validated.</td>
<td>42</td>
</tr>
<tr>
<td>4.2</td>
<td>Top class structure of DOOP. Some classes and individuals are not shown due to space limitations</td>
<td>53</td>
</tr>
<tr>
<td>4.3</td>
<td>Example of individuals in DOOP. Some classes and individuals are not shown due to space limitations</td>
<td>53</td>
</tr>
<tr>
<td>4.4</td>
<td>Mapping DOOP categories to OPP-115</td>
<td>56</td>
</tr>
<tr>
<td>4.5</td>
<td>Privacy policy page on Tanger Outlets’ website uses a dropdown, hence was not correctly scraped</td>
<td>59</td>
</tr>
<tr>
<td>4.6</td>
<td>Privacy policies: sentences versus words.</td>
<td>60</td>
</tr>
<tr>
<td>4.7</td>
<td>Privacy policies: average lengths of privacy policies.</td>
<td>60</td>
</tr>
<tr>
<td>4.8</td>
<td>Sentences selected for reading for query 1: ‘Does this website share my personal information with third-parties?’</td>
<td>61</td>
</tr>
<tr>
<td>4.9</td>
<td>Sentences selected for reading for query 2: ‘Does this website use tracking cookies?’</td>
<td>62</td>
</tr>
<tr>
<td>4.10</td>
<td>Sentences selected for reading for query 3: ‘Can I opt-in/opt-out of information gathering?’</td>
<td>63</td>
</tr>
<tr>
<td>4.11</td>
<td>Results for experiment 2 for query 2 after adding ‘cookie’ to the keywords</td>
<td>66</td>
</tr>
<tr>
<td>A.1</td>
<td>Sentences selected for reading for query 1: ‘Does this website share my personal information with third-parties?’</td>
<td>83</td>
</tr>
<tr>
<td>A.2</td>
<td>Sentences selected for reading for query 1: ‘Does this website share my personal information with third-parties?’</td>
<td>84</td>
</tr>
<tr>
<td>A.3</td>
<td>Sentences selected for reading for query 2: ‘Does this website use tracking cookies?’</td>
<td>85</td>
</tr>
<tr>
<td>A.4</td>
<td>Sentences selected for reading for query 2: ‘Does this website use tracking cookies?’</td>
<td>86</td>
</tr>
<tr>
<td>A.5</td>
<td>Sentences selected for reading for query 3: ‘Can I opt-in/opt-out of information gathering?’</td>
<td>87</td>
</tr>
<tr>
<td>A.6</td>
<td>Sentences selected for reading for query 3: ‘Can I opt-in/opt-out of information gathering?’</td>
<td>88</td>
</tr>
<tr>
<td>A.7</td>
<td>Results for experiment 2 for query 2 after adding ‘cookie’ to the keywords.</td>
<td>89</td>
</tr>
<tr>
<td>A.8</td>
<td>Results for experiment 2 for query 2 after adding ‘cookie’ to the keywords.</td>
<td>90</td>
</tr>
</tbody>
</table>
List of Listings

4.1 SPARQL query for returning terms from DOOP related to personal information and third-parties. .................................................. 48
4.2 SPARQL query for returning terms from DOOP that are related to tracking and cookies ............................................................................. 50
4.3 SPARQL query for DOOP to return terms related to opt-in and opt-out ....................................................................................... 50
## List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMU</td>
<td>Carnegie Mellon University</td>
</tr>
<tr>
<td>CSV</td>
<td>Comma Separated Values</td>
</tr>
<tr>
<td>COPPA</td>
<td>Children’s Online Privacy Protection Act</td>
</tr>
<tr>
<td>CQ</td>
<td>Competency Question</td>
</tr>
<tr>
<td>DF</td>
<td>Document Frequency</td>
</tr>
<tr>
<td>DOOP</td>
<td>Domain Ontology for Online Privacy</td>
</tr>
<tr>
<td>FIPPs</td>
<td>Fair Information Practice Principles</td>
</tr>
<tr>
<td>IEEE</td>
<td>Institute of Electrical and Electronics Engineers</td>
</tr>
<tr>
<td>IR</td>
<td>Information Retrieval</td>
</tr>
<tr>
<td>KM</td>
<td>Knowledge Management</td>
</tr>
<tr>
<td>k-NN</td>
<td>k-Nearest Neighbour</td>
</tr>
<tr>
<td>MOU</td>
<td>Memorandum Of Understanding</td>
</tr>
<tr>
<td>NB</td>
<td>Naive Bayes</td>
</tr>
<tr>
<td>NLP</td>
<td>Natural Language Processing</td>
</tr>
<tr>
<td>OECD</td>
<td>Organization for Economic Co-operation and Development</td>
</tr>
<tr>
<td>ORSD</td>
<td>Ontology Requirements Specification Document</td>
</tr>
<tr>
<td>OWL</td>
<td>Web Ontology Language</td>
</tr>
<tr>
<td>P3P</td>
<td>Platform for Privacy Preferences</td>
</tr>
<tr>
<td>PET</td>
<td>Privacy Enhancing Technology</td>
</tr>
<tr>
<td>PII</td>
<td>Personally Identifiable Information</td>
</tr>
<tr>
<td>POS</td>
<td>Part-Of-Speech</td>
</tr>
<tr>
<td>RAKE</td>
<td>Rapid Automatic Keyword Extraction</td>
</tr>
<tr>
<td>RDF</td>
<td>Resource Description Framework</td>
</tr>
<tr>
<td>SPARQL</td>
<td>SPARQL Protocol And RDF Query Language</td>
</tr>
<tr>
<td>ST</td>
<td>Semantic Technology</td>
</tr>
<tr>
<td>SVM</td>
<td>Support Vector Machine</td>
</tr>
<tr>
<td>TF-IDF</td>
<td>Term Frequency - Inverse Document Frequency</td>
</tr>
<tr>
<td>ToS;DR</td>
<td>Terms of Service; Didn’t Read</td>
</tr>
<tr>
<td>URI</td>
<td>Unique Resource Identifiers</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

The Internet has become an integral part of our lives. It has penetrated all sectors of our daily activities: business, communication, shopping, and personal life. Many of these activities place the cost of privacy on the users by requiring them to disclose their personally identifiable information (PII) in exchange of seemingly ‘free’ services. Due to the near permanent nature of the Internet, this loss of privacy can have a long-lasting effect on the user. A 2015 Pew Research Centre survey found that 91% of American adults either agree or strongly agree that they have lost control of how their private information is collected and used (Madden & Rainie, 2015). The collection of PII by online service providers is often justified with claims of creating a more user-centric web experience. However, PII is sold and shared frequently with third parties that use it to profile users and track them across domains. While users are increasingly concerned about their privacy online (Jensen & Potts, 2004) they scarcely understand the implications of PII sharing (Winkler & Zeadally, 2016). To ease user concerns and bolster trust, companies are introducing privacy enhancing technologies (PET) such as: opt-out mechanisms; reducing the amount of personal information collected; anonymization of personal data; and ‘layered’ policies (Ten steps to develop a multilayered privacy notice, 2006; Munur, Branam, & Mrkobrad, 2012). Without these becoming a common standard, opaque and verbose policies are still the norm. In the mean time, it would help the users if there existed a context aware system for real-time privacy policy analysis that parses policies and identifies information relevant to the user’s concerns.

Domain ontologies can be used to not only structure concepts but also describe relationships between them. Used in concert with natural language processing (NLP) algorithms, they can help users sort through long policies and find pertinent information, and thus increase transparency and understanding of privacy policies.

As a first step in helping users find pertinent information in privacy policies, this thesis presents the development of a domain ontology using NLP algorithms as a way to reduce costs and speed up development.

The rest of this Chapter is organized as follows:
1. In Section 1.1, background on privacy policies is provided along with their importance, and their usability.

2. Section 1.2 provides a brief introduction to ontologies, including: what they are; how they are built; and how they are validated.

3. Section 1.3 briefly introduces the problems associated with text annotations. An issue that concerns most of the research in this thesis.

4. Section 1.4 presents a general view of the problem tackled in this thesis, the approach taken to solving it, and the validation criteria.

1.1 Privacy Policies

When online service providers (e.g., websites) collect, store, and disseminate personal information; or generate, store, and disseminate meta-data based on collected information of their users without their explicit permission the user loses 'ownership' of that data and their privacy ([Friedman, Khan Jr, & Howe 2000][Taddicken & Jers 2011]). Privacy polices offer a glimpse into how users data is being collected and disseminated. They are designed to reduce fear among users concerning their personal information ([Westin 1967]). By law, privacy policies are required to disclose the nature and extent of information collection ([Privacy Online: Fair Information Practices in the Electronic Marketplace 2000][REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation), 2016][Personal Information Protection and Electronic Documents Act 2000][Digital Privacy Act 2015]). Unfortunately, most policies are often lengthy, difficult and time-consuming to read, and as a result are infrequently read ([McDonald & Cranor 2008][Mcdonald, Reeder, Kelley, & Cranor 2009][Milne & Culnan 2004][Jensen & Potts 2004]). The demotivating nature and the difficulty of reading privacy policies amounts to a lack of transparency. Failing to provide usable privacy policies prevents users from making informed decisions and can lead them to accept terms of use jeopardizing their privacy and PII. Recently, Cranor et al. showed through their analysis of 75 policies that most policies do not provide enough transparency about data collection for the users to make informed privacy decisions ([Cranor, Hoke, Leon, & Aul 2015]).

In addition to length and readability, privacy policies also differ from one another by their content of legal and technical jargon, and coverage ([Mcdonald et al. 2009][McDonald & Cranor 2008][Wilson, Schaub, Dara, et al. 2016]). While it is true that FIPPs (Fair Information Practice Principles) and OECD (Organization for Economic Co-operation and Development) offer general guidelines for writing privacy policies, they only provide a conceptual framework; a qualitative review of policies reveals that language and structure being used differs between policies and economic zones (E.U., U.S.A, Canada) ([Mcdonald et al. 2009][Sadeh et al. 2013]). There is also an inconsistent amount of jargon used between policies, and policies of organizations in the E.U. tend to have supplementary
information that tends to be absent from American and Canadian policies (Wilson, Schaub, Dara, et al., 2016; Sun, 2012; Cranor et al., 2015). Boilerplate language is mostly the norm for cookie policies.

To aid users, several attempts have been made to simplify policies. The most notable of these efforts was the Platform for Privacy Preferences (P3P) (Cranor, Langheinrich, & Marchiori, 2002; Cranor, 2003), which introduced a machine readable format for creating privacy policies. This intent was that the standardized format would make it easier to extract relevant information with the help of logic systems, e.g., reasoners. Unfortunately, P3P had limited success due to a lack of industry and developer participation. It also lacked proper policy validation which prevented policy developers from creating accurate policies (Lämmel & Pek, 2013).

In a previous study, Wilson et al. (2016) created the OPP-115 corpus, a corpus of 115 manually annotated privacy policies with 23,000 data practices. A data practice is roughly defined as a purpose or consequence of collecting, storing, or generating data about a user. In the study, the 10 domain experts (privacy experts, public policy experts, and legal scholars) used a custom designed web-based tool to annotate practices and assign various attributes to them for classification purposes. Each policy took an average of 72 minutes to annotate. Whilst this approach produced a high quality and nuanced data set, it is costly to expand and maintain such a knowledge base both in time and money.

In a different approach, Guntamukkala et al. (2015) used machine learning algorithms to measure completeness of privacy policies. To do this, they first identified eight important sections that privacy policies must have to be considered ‘transparent’, based on guidelines presented under Organization for Economic Co-operation and Development (OECD) and United States Federal Trade Commission’s (FTC) Fair Information Practice Principles (FIPPs). Then classifiers were used to classify bi-grams extracted from 100 manually annotated privacy policies into the eight sections. Their results showed that all eight categories achieved a completeness score of over 75%. While their data set only considered the most visited websites, it is a promising result that requires more investigation.

Crowdsourcing is a form of distributed work where a task is broken down into smaller micro-tasks which are completed by a host of online workers; the result is the aggregated efforts of all of the workers. The micro-tasks usually require little to no training, hence reducing the overall cost of completing the task. Automation and crowdsourcing can reduce the cost of creating and maintaining such a data set, and still maintain reasonable quality. Terms of Service; Didn’t Read (ToS;DR) (Terms of Service; Didn’t Read (ToS;DR), n.d.) is a project that uses crowdsourced annotations to answer key questions about the policies. The limiting factor with crowdsourcing is the large scale participation rate required to ensure success, leading to delays in the success of the project. To remedy this, researchers (Zimmeck & Bellovin, 2014; Sadeh et al., 2013) tried to combine ToS;DR, natural language processing (NLP) and other machine learning techniques with the goal of automatically inferring privacy concerns from privacy policies. Whilst these techniques work well in recognizing the pre-determined classes of privacy concerns, they still rely on quality, reliable, and up-to-date crowdsourced data which is presently lacking due to the inherent de-motivating nature of reading privacy policies for completing crowdsourced task
In the research conducted by Ramnath et al., the researchers proposed combining machine learning and crowdsourcing (for validation) to semi-automate the extraction of key privacy practices (Ramnath et al., 2014). Through their preliminary study they were able to show that non-domain experts were able to find an answer to their privacy concern relatively quickly (≈ 45s per question) when they were only shown relevant paragraphs that were mostly likely to contain an answer to the question. They also found that answers to privacy concerns were usually concentrated rather than scattered all over the policy. This is an important find because it means that if users were directed to relevant sections in the policy they would be able to address their privacy concerns relatively quickly instead of reading the entire policy. Additionally, Pan and Zinkhan have showed that when users are presented with a short and straightforward policy, they are more inclined to read it (Pan & Zinkhan, 2006).

In a more recent user study conducted by Wilson et al. (2016), the quality of crowdsourced answering of privacy concerns was tested against domain experts with particular emphasis on highlighted text. The researchers found that highlighting relevant text had no negative impact on accuracy of answers. They also found out that users tend not to be biased by the highlights and are still likely to read the surrounding text to gain context and answer privacy concerning questions. They also found an 80% agreement rate between the crowdsourced workers and the domain experts for the same questions (Wilson, Schaub, Ramnath, et al., 2016). Similarly, Mysore Sathyendra et al. showed through their study that it was possible to highlight and extract opt-out practices from privacy policy using keywords and classification algorithms with reasonable accuracy; of the various models tested, best model used a logistic regression classification algorithm with a manually crafted feature set and achieved an F1 score of 59% (Sathyendra, Schaub, Wilson, & Sadeh, 2016).

The general drawback of crowdsourcing, especially with respect to privacy policies, is that it relies on non-expert users to read policies to provide data. Since most users are not motivated to reading policies to begin with, it would take a long time to crowdsource enough data to be useful. However, what is clear is that highlighting relevant text with appropriate keywords can still provide some feedback to the concerned users that are inclined to read shorter policies. One way to automatically highlight relevant text in privacy policies, in a manner that can be easily scaled, is by using Semantic Technologies (ST) such as an ontology (Section 1.2).

PrivOnto is an ontology created by Oltramari et al. (2016) with an intent to analyze privacy policies. It was constructed using the OPP-115 policies data set created by (Wilson, Schaub, Dara, et al., 2016). Since the ontology captures all of the detailed annotations created by the domain experts, it serves as an excellent tool to understand how the policies are constructed. The analysis can only be conducted on policies that have been instantiated into the ontology. This, in turn, means that any new policy that needs to be added to the ontology must first be annotated by one or more domain experts. Since each policy takes about 72 minutes to annotate by a domain expert, the per hour rate of hiring this expert makes this a very expensive process; especially, if one wishes to add enough policies into the ontology to make it useful for a large number of sites. As such, PrivOnto remains
an academic and policy-making tool rather than a commercial one.

1.2 Ontologies

Ontologies capture domain specific taxonomy and preserve context; which can be used in conjunction with a reasoner to reason over a particular document providing useful insights. Historically, they have been widely used in the domains of healthcare, animal and plant biology (Fridman Noy & Hafner 1997), but only limited use in a legal or semi-legal domain. A key component of an ontology are its objects, usually represented as a taxonomy which is constructed by extracting keywords from the corpus of the targeted domain (privacy policies in this case) (Chandrasekaran, Josephson, & Benjamins 1999). Usually, domain experts are employed to aid with creating ontologies and highlighting key concepts that need to be part of the taxonomy.

1.2.1 What is an Ontology?

Though several definitions exist, the most quoted definition for what an ontology is given by Thomas R. Gruber (1993), “a formal, explicit specification of a shared conceptualization”. A conceptualization, in this case, is the model abstraction of some real-world phenomena usually restricted to some domain. The explicit specification stems from the well named and well defined concepts that are being used, along with their respective constraints (relationships). A name is a term, and its definition explains how it relates to another. Formal meaning that the knowledge is represented in a mathematically sound language, usually logic based, whose axioms and properties are well established. The implication of formality here is that the encoded language can be coded into a computer and be used to derive more information through inference. The formality also removes any ambiguity that is often introduced when knowledge is encoded in a natural language. Finally, shared means that the knowledge is universally accepted within the target domain’s group and can be reused not only within the group but also exchanged with groups of other domains (Corcho, Fernández-López, & Gómez-Pérez, 2003).

It is generally accepted that ontologies have two basic features (Uschold & Gruninger 2004):

1. A taxonomy of terms used to name and describe the objects (concepts) being described by the ontology.

2. A specification, grounded in logic, used to add meaning between terms.

Ontologies may describe a wide variety of things in a domain, but they all share a common set of attributes (Bermejo 2007):

- **Classes** capture the core vocabulary that is used to describe a domain. They are also referred to as concepts, and are generally arranged in a hierarchical or taxonomical form as classes and sub-classes.
• **Relations** are definitions of how concepts are inter-related to one another.

• **Attributes** are the properties associated with classes that describe the features of that class.

• **Formal axioms** are logical statements that always evaluate to true.

• **Functions** are a special case of relations.

• **Instances** are elements of a class; and are also called *individuals*. Not all ontologies must have these; but if they do then that ontology constitutes a *knowledge base*.

There are many different types of ontologies that differ based on not only their purpose but also their content. The purpose of the ontology is determined by how widely it is meant to be used. A general world knowledge ontology could describe a language or everyday things, typically a commonsense knowledge domain. Versus a domain specific ontology that captures concepts, relations, and attributes belonging to a narrow field of interest (Fridman Noy & Hafner, 1997). The content of the ontology is determined by the richness of the term definitions. On one extreme, there could be an ontology of only a taxonomy of terms, without any attributes, which capture the vocabulary used in a domain. On the other extreme, the ontology could be very well defined and thoroughly capture the target theory (domain or subject) where all the concepts, relations, and attributes have been exhaustively formalized (Uschold & Gruninger, 2004).

### 1.2.2 Building Ontologies

Just as there are a variety of ontologies which differ based on their purpose and content, just as many methodologies exist to create them. There exists two main methods of ontology creation; one involving only skilled ontology engineers, and the other involving ontology engineers and domain experts. The goal of the ontology determines the method. Hence, requirements engineering forms the first and most important step of creating an ontology. When it comes to constructing the ontology, majority of the methods do not involve domain experts (Slimani, 2014). The general steps to develop an ontology are:

- Define classes
- Organize classes hierarchically in a taxonomy
- Define relationships between classes
- Define class attributes and properties
- Add instances for the classes
- If there exists sufficient knowledge and time then create axioms and functions

Building ontologies are generally considered an iterative process. Thus, a lightweight ontology with only a well defined domain taxonomy is still considered useful.
1.2.3 Evaluating Ontologies

Evaluating an ontology is difficult, and due to the relative infancy of the field of ontology learning, not many frequently used techniques exist to evaluate ontologies (Biemann, 2005). There are a few ways of evaluating an ontology, each depending on the type and purpose of the ontology constructed. Due to the complexity of the ontology, they are generally evaluated on a level by level basis. This is a good strategy because each ontology may be constructed in a different manner resulting in a varied number of physical characteristics: clusters, hierarchies, relationships, density of taxonomy at ever level, number of terms, and depth (Fridman Noy & Hafner, 1997; Brank, Grobelnik, & Mladenić, 2005). A few of the techniques of evaluating the complete ontology are described below:

Formal competency questions (CQs)

In this evaluation strategy proposed by (Grüninger & Fox, 1995; Fridman Noy & Hafner, 1997), informal competency questions (queries) are first expressed formally. These questions are requirements that are in the form of questions that an ontology must be able to answer. The formal questions are then evaluated using completeness theorems with respect to first (axioms) and second order (situational calculus) logic representation of concepts, attributes, and relations.

Practical applications

Most popular form of evaluation, involves creating an application that the ontology was designed for, demonstrating conceptual coverage and practical usefulness (Fridman Noy & Hafner, 1997; Brank et al., 2005; Biemann, 2005). A lot of times application aren’t necessarily built but simply envisioned (Fridman Noy & Hafner, 1997).

Gold standard

One of the simplest ways of evaluating an ontology is to compare it with an existing ontology or a “gold standard” and evaluating the improvements in the new one (Brank et al., 2005; Maedche & Staab, 2002; Biemann, 2005).

Data driven

At times when a “golden standard” does not exist, ontologies are simply compared with the sources of data from the domain that the ontology is meant to cover. This involves statistically extracting key terms and concepts from the corpus the ontology is meant to cover and evaluating if they exist in the ontology itself. This is done via calculation of the precision and recall (Section 1.4.3) scores (Brank et al., 2005; Liu, Hogan, & Crowley, 2011; Velardi, Fabriani, & Missikoff, 2001a; Naik, 2015). To allow for synonyms, etc,
instead of the lexical term themselves, hypernyms from WordNet for those words can be used instead ([Brewster, Alani, Dasmahapatra, & Wilks 2004]).

Domain experts

In this case the resultant ontology is evaluated by a human expert who tries to assess how well the ontology meets a set of predefined criteria, standards, and requirements ([Brank et al. 2005], [Liu et al. 2011]).

1.3 Taxonomy Generation

In order to build a domain ontology, the vocabulary of the domain needs to be captured and organized into a taxonomy. There are two main ways to capture vocabularies: manually and automatically. Manually capturing the vocabulary involves reading and knowing domain text which would involve hours of manual labour that can be costly. An easier approach would be to use NLP techniques to automatically extract keywords for taxonomy creation ([Cimiano & Völker 2005], [Velardi, Fabriani, & Missikoff 2001b]). This requires fewer man hours compared to the manual methods; hence, is cheaper. Issues related to NLP, and more specifically keyword extraction, are discussed in this section.

The following is divided into three parts:

1. Keyword and keyphrase extraction.
2. Approaches to keyword extraction.
3. Text pre-processing.

1.3.1 Keyword & Keyphrase Extraction

Automatic keyphrase extraction algorithms have been evaluated over a diverse set of corpora ranging from scientific papers to twitter and email messages. There are several factors that affect the performance of automated key term (keyword) extraction algorithms: length, structural consistency, topic change, and topic correlation ([Hasan & Ng 2014]), as described below. The key terms are usually short-listed from a list of candidate terms based on some metric (Section 1.3.2).

Length

It is often difficult to determine and select the right number of key terms from a longer (> 1000 words ([Lv & Zhai 2011])) document; the longer an input document, the greater the number of candidate terms to choose from. For example, scientific papers can have hundreds of candidate terms in them with at least 10 key terms being selected per paper
Structural Consistency

Key terms in structured documents like scientific papers, tend to be clustered around certain areas such as the abstract, introduction, and title [Kim & Kan, 2009]. Similarly, structural consistency facilitates key term extraction in web pages with metadata [Yih, Goodman, & Carvalho, 2006] and dialogue acts in online chats [Kim, Baldwin, & Kan, 2010]. Thus, for any textual data set lacking in standard structure this property cannot be used.

Topic Change

In a body of text, the topic being discussed might stay constant throughout the document (e.g. scientific papers and news articles) or it might change as the text progresses (e.g. conversations, chats, and email). As such, keyphrases found at the beginning and end of the text might be consistent [Witten, Paynter, Frank, Gutwin, & Nevill-Manning, 1999; Medelyan, Frank, & Witten, 2009], or continue to change as the topic being discussed changes [Kim & Baldwin, 2012]. Furthermore, detection of topic change is a difficult task to automate if the textual data set is unstructured (e.g., blogs or web pages) unless there exists a summary or index at the beginning of the document (e.g. meeting notes).

Topic Correlation

A common observation exploited during keyword extraction is the relatedness of keyphrases in a document. This aids in recognizing neighbourhoods (cluster of candidate terms) from which keywords can be ranked in terms of ‘weights’ within the cluster [Mihalcea & Tarau, 2004; Turney, 2003]. The resulting keywords become the desired keyword for the document. This is however difficult to do in text documents where the topics are constantly changing (e.g. chats, personal emails, informal meetings, personal blogs). In such texts, it is difficult to form topic related neighbourhoods of candidate terms to select key terms for the document.

1.3.2 Approaches to Keywords Extraction

Overall, there are four main computational approaches to keyword extraction: rule based linguistic, statistical, machine learning, and domain specific [Siddiqi & Sharan, 2015].
Rule Based Linguistic

This approach includes lexical, syntactic, and discourse analysis that utilizes the linguistic features and knowledge of the text. This approach requires extensive domain knowledge as well as language expertise. Despite yielding superior keywords, the task of forming enough rules to guarantee success is a manual and expensive process.

Statistical

This approach is based on the features derived from the statistical analysis of a corpus, e.g. document term frequency, corpus term frequency, relative position of the first term occurrence, amongst others. As such, this approach is language independent. The overall yield of keywords is not as good as the Rule Based Linguistic approach, but this shortcoming is compensated for by the ability to perform keyword extraction on a greater diversity of data sets and achieve satisfactory results (Sarawagi, 2008).

Machine Learning

Machine learning can be divided into two broad types: supervised and unsupervised learning. The majority of the machine learning techniques involve supervised learning algorithms which rely on a tagged corpus for training a model to learn features (keywords) from the text. After sufficient training, the model is then applied on similar corpus to extract keywords. The keyword assignments made over the training data set forms the reference, also known as controlled vocabulary, and treated as classes used in a classification problem. Some examples of supervised learning algorithms include, K-nearest neighbour (k-NN) (Yeung, Gibbins, & Shadbolt, 2008; Tan, 2005), Naive Bayes (NB) (Frank, Pynte-ter, Witten, Gutwin, & Nevill-Manning, 1999), GenEx (Turney, 2000), and Support Vector Machines (SVM) (Zhang, Xu, Tang, & Li, 2006).

Since creating a tagged corpus is a very time consuming task, unsupervised learning algorithms are used which do not require any training set for the training of models. They instead rely on linguistic and statistical features of the text. The task is framed as a ranking or clustering problem.

Domain Specific

If the structure of the documents in a given corpus is known, and the lexical structure of the keywords in the domain of that corpus is known, then a set of hybrid methods can be used to extract keywords from it. It need not be as work intensive as the Rule Based Linguistic approach.
1.3.3 Text Pre-Processing

An important step in Information Retrieval (IR) is candidate phrase selection. The candidate phrase selection typically involves text pre-processing steps such as: normalization, filtration, stop word removal, removal of punctuation, stemming, part-of-speech (POS) tagging, tokenization, and n-gram filtration. Pre-processing can significantly improve evaluation metrics as it gets rid of irrelevant text within the text and reduces the search space or the amount of text that needs processing ([Wang, Liu, & McDonald, 2014]). A brief description of each is given below.

Tokenization

Tokenization is the act of breaking up a character string (generally sentences) into pieces, called tokens, sometimes throwing away certain characters within them, such as punctuation, e.g. 555-555-5555 becomes 5555555555, and O’Henry becomes OHenry. A token is the smallest indivisible semantic unit, that is a sequence of characters grouped together to be used for further processing. They are also sometimes referred to as words or terms, but could also mean a uniform character sequence, e.g. telephone numbers, or ISBN numbers ([Manning, Raghavan, Schütze, et al. 2008]).

Normalization

Normalization is the process where a token is replaced with an equivalent token, where the superficial differences between their character sequences are removed. For example, ‘U.S.A’ is changed to ‘USA’, and ‘anti-inflammatory’ is changed to ‘antiinflammatory’. Manual rules can also be created to swap terms, such ‘an aircraft’ instead of ‘an aeroplane’. Removing of accents and diacritics is also a type of normalization; as is case-folding where all of the tokens in the document are changed to a lower case ([Manning et al. 2008]). Localization is another case of normalization where words with spelling from other dialects of a language are replaced with the corpus’ dialect to ensure consistency. For example, color (American English) becomes colour (British/International English), and torch British/International English becomes flashlight (American English).

Stop Word Removal

Words that are very common and that appear to offer little value are usually removed from the vocabulary. These words are called stop words. Generally, stop words are manually selected based upon their document frequency, where the most frequent terms are discarded upon review and stored in a stop list for further processing ([Manning et al. 2008]). Alternatively, general use stop lists based on lexical analysis using a part-of-speech (POS) tagger can also be used. These lists are usually used to remove the stop words from the document, e.g. auxiliary verbs, conjunction, and articles.
Table 1.1: Comparing the results of stemming using Porter & Lovin algorithms.

<table>
<thead>
<tr>
<th>Unstemmed Word</th>
<th>Lovins</th>
<th>Porter</th>
</tr>
</thead>
<tbody>
<tr>
<td>superconduct(ed/ing)</td>
<td>superconduc</td>
<td>superconduct</td>
</tr>
<tr>
<td>superconduction</td>
<td>superconduc</td>
<td>superconduct</td>
</tr>
<tr>
<td>superconductive</td>
<td>superconduc</td>
<td>superconduct</td>
</tr>
<tr>
<td>superconductively</td>
<td>superconduc</td>
<td>superconductiveli</td>
</tr>
<tr>
<td>superconductivity(’s)</td>
<td>superconduc</td>
<td>superconductiviti</td>
</tr>
<tr>
<td>superconductor(s/’s)</td>
<td>superconduc</td>
<td>superconductor</td>
</tr>
</tbody>
</table>

Stemming

Stemming is the process of reducing a word to its root form by chopping off its end. For instance, ‘collecting’, ‘collection’, ‘collected’, and ‘collects’, are all derived from the common base word ‘collect’. Stemming algorithms utilize heuristics to achieve this effect and are not always right but are close enough for most words. A comparison between the Porter’s (Porter, 1980) and Lovin’s (Lovins, 1968) stemming algorithms is shown in Table 1.1 (Hull et al., 1996). It is clear from the example that Lovin’s algorithm correctly stems all the words to a common root; as the goal here is to reduce unnecessary variance in the word forms so that a higher number word matches can be achieved. But unlike this example, Porter’s stemming algorithm is the most commonly used stemming algorithm for English due to its repeatedly demonstrated high efficacy (Manning et al., 2008).

N-gram Filtration

A contiguous sequence of $n$ tokens is called an $n$-gram. A ‘gram’ can be interpreted as a token, and as previously discussed (Section 1.3.3) a token can be a sequence of numbers and alphanumeric characters. Hence, a 1-gram or unigram is a single token, a 2-gram or bigram is a sequence of two tokens, a 3-gram or trigram is a sequence of 3 tokens, and so on. Deciding on which n-grams to exclude involves empirical qualitative analysis dependent on the corpus being analyzed.

1.4 Methodology

This section provides an overview of the central problem which this body of research is trying to solve, and outlines the proposed solution. It is broken up into two main sections: section 1.4.1 discusses the central problem which forms the theme of this thesis, and section 1.4.2 covers the progression of the solution.
1.4.1 Problem Statement

As established in Section 1.1, online privacy policies remain unusable to the average users due to their length and elusive language. This contributes to a lack of transparency which in turn leads to uninformed decisions and risk-averse behaviour. Policies that are usable tend to be read more often and give the users more confidence in sharing their personal information. Which means that a usable privacy benefits all parties involved. Previous attempts at making privacy policies more usable by voluntary industry-wide adoption of structured policies was met with limited success. Research shows that policies which highlight sections that directly address the user’s concerns tend to be read more often, as it reduces the cost of reading the policies. Hence, a solution is required which considers concerns of all stakeholders: the online service providers that create privacy policies, as well as the users that do not necessary like reading them. This solution must not require the online service providers to change their policies drastically, hence avoiding push-back, but must reduce the amount of text that the users have to read, and direct users to the text that pertains directly with their concerns.

1.4.2 Proposed Solution

In order to direct users to the relevant text within the policies, there needs to be a way to evaluate the text and highlight all relevant sections. Since the language being used within the policies differs so greatly [McDonald et al. 2009; Sadeh et al. 2013], there needed to be a system that is able to capture all the variations of a topic. Semantic technologies such as NLP and ontologies are two well-known ways of mining text and reasoning. Since domain ontologies can capture the vocabulary of a domain and specify rules about each term, it is possible to capture the diversity of concepts within online privacy domain and reason over them to find equivalent terms for analysis. Through NLP, it is possible to logically break apart the text in a policy, and working in conjunction with the ontology, be able to recognize important sections within the policies. Since there are several open-source, free, and well-used NLP libraries, what is required is an ontology that is: easy to build, low maintenance, and relatively inexpensive to scale. This thesis proposes such an ontology.

1.4.3 Validation

As it is traditionally done with IR, the results from each of the keyword extraction algorithms were compared using 3 main scores: precision, recall, f-measure. Precision and recall are well-known evaluators of IR proposed by [Salton & Buckley 1988]. Jaccard’s similarity co-efficient between the manual annotation and each of the resultant sets to understand their similarity were also calculated.
Precision:

*Precision* ($P$) is the ratio of the number of relevant terms returned from a term extraction algorithm ($\{\text{manually-selected}\} \cap \{\text{machine-selected}\}$) to the total numbers of retrieved terms by the algorithm ($\{\text{machine-selected}\}$). The precision is calculated using Equation 1.1.

\[
P = \frac{|\{\text{manually selected}\} \cap \{\text{machine-selected}\}|}{|\{\text{machine-selected}\}|} \quad (1.1)
\]

Recall:

The *recall* ($R$) of an information system is defined as the ratio of the number of relevant terms returned to the total number of relevant terms in the collection. The recall is computed using Equation 1.2.

\[
R = \frac{|\{\text{manually selected}\} \cap \{\text{machine-selected}\}|}{|\{\text{manually selected}\}|} \quad (1.2)
\]

F-measure:

The *F-measure* or $F_1$-score is a special case of $F_\beta$-score where the discriminatory $\beta$ value is set to 1.0. *F-measure* is a metric that combines precision and recall into a single number (Van Rijsbergen [1979]). It is calculated using Equation 1.3 where $\beta$ value is the relative importance given to the precision score over recall. For the task of keyword and keyphrase extraction from online privacy policies, both precision and recall were deemed equally important. Hence, a $\beta$ value of 1.0 or *F-measure* was chosen as shown in Equation 1.4.

\[
F_\beta\text{-score} = \frac{(\beta^2 + 1.0) \cdot \text{precision} \cdot \text{recall}}{\beta^2 \cdot \text{precision} + \text{recall}} \quad (1.3)
\]

\[
F_1\text{-score or F-measure} = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (1.4)
\]

Jaccard’s Similarity Co-efficient:

Jaccard index is a measure often used for comparing similarity, dissimilarity, and distance between data sets. Measuring the Jaccard similarity coefficient between two data sets is the result of dividing the number of terms that are common to both sets and the total number of terms. It is illustrated in the Equation 1.5.

\[
J(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (1.5)
\]
Ontology validation

The ontology was validated using a data driven approach against the OPP-115 dataset created by the Carnegie Mellon University’s Usable Privacy research group (Wilson, Schau, Dara, et al., 2016).

1.4.4 Contribution

The primary contributions of this body of research are three-fold:

1. In order to gather keywords for the taxonomy for online privacy, test the efficacy of unsupervised and supervised learning algorithms for extracting keywords and keyphrases from privacy policies.

2. To improve the quality of keywords extracted by supervised learning algorithms by creating a richer training data set while keeping the costs low, examine the difficulty of training non-domain experts for the task of extracting keywords and keyphrases and compare the results against the results from a domain expert.

3. Using the top 3 identified privacy-related concerns from users, building an ontology that would return keywords that would highlight sections of privacy policies that would help the users address those concerns. The ontology also categorizes keywords and queries, so that it can be used as a recommendation system.
1.4.5 Thesis Organization

Figure 1.1 shows the general workflow of different steps taken to create and validate a domain ontology for the online privacy. The rest of the thesis is organized as such:

- Chapter 2 presents how the baseline for keywords was created. Results from unsupervised learning algorithms for keyword extraction from online privacy policies are then shared and analyzed.

- Due to the time consuming nature of creating baselines and “gold standards” by domain experts, it was important to investigate how trained non-domain experts would fare. This is explored in Chapter 3. A supervised learning algorithm for keyword extraction is also investigated, and analyzed against the baseline and the results from the best faring unsupervised learning algorithm.

- Chapter 4 presents the domain ontology for online privacy, along with its validation against CMU’s OPP-115. Usability of the privacy policies using the ontology is also demonstrated.
Chapter 2

Extracting Keywords From Online Privacy Policies With Unsupervised Learning Algorithms

As previously noted in Section 1.2.1, there are two basic parts of an ontology: the taxonomy, and the logic about and between the taxonomical terms. In a domain ontology, the taxonomy captures the vocabulary of the domain. As such, domain experts are typically needed to manually construct the ontology. NLP techniques have been employed to extract keywords for the taxonomy creation (Velardi et al., 2001b; Cimiano & Völker, 2005). For a domain as large as online privacy, manual creation of the taxonomy can be a very expensive pursuit. As such, automatic keyword extraction techniques were employed to enrich our data set of manually extracted keywords to be used to create a taxonomy for the online privacy domain.

Keyword extraction algorithms such as Rapid Automatic Keyword Extraction (RAKE) (Rose, Engel, Cramer, & Cowley, 2010) and TextRank (Mihalcea & Tarau, 2004) have shown promising results over traditional benchmarks such as the Term Frequency-Inverse Document Frequency (TF-IDF). They employ statistical learning over a custom training set by creating an array of word tokens or clusters of words, which are then ranked based on the classification of whether they are positive or negative examples of keyphrases. The classifier takes into consideration word frequency in a document, surrounding stopwords, and term clusters of words with similar meaning, as important features in a document that determine the relevance of terms. These algorithms have also been previously used to extract keywords and keyphrases (multiple keywords that occur together) from legal documents (Jungiewicz & Łopuszyński, 2014; Le, Shirai, Le Nguyen, & Shimazu, 2015).

To the best of our knowledge, no attempt has been made to extract keywords from privacy policies or English legal documents. However, there were attempts to extract keywords from Polish and Japanese legal documents. In the research conducted by Jungiewicz et al. (Jungiewicz & Łopuszyński, 2014), the Polish researchers used RAKE algorithm with a custom built stoplist (list of stop words) of Polish words. Input consisted of 11,000
rulings from the National Appeals Chamber from the Polish Public Procurement Office.
The results were qualitatively compared with results from running RAKE with a standard
information retrieval stoplist. The researchers concluded that their custom stoplist worked
better for the sample size that they analyzed, however, their results remain.

In a similar approach, Japanese researchers (Le et al., 2015) proposed an algorithm using
stop words as delimiters to select candidate keywords: single keywords as well as keyphrases.
The algorithm is very similar to RAKE, but uses Okapi BM25 (Robertson et al., 1995)
ranking function and Term Frequency - Inverse Document Frequency (TF-IDF) to cal-
culate candidate keyword’s scores. The researchers benchmarked their algorithm against
a manually annotated Japanese National Pension Act and compared their results with a
similar implementation using the TextRank keyword extraction algorithm. They concluded
that their algorithm achieved an F1-score of 8% higher than TextRank. Notwithstanding
these efforts, keyword extraction in the realm of legal documents remains an elusive task.

One of the problems that arises in keyword extraction is that of topic correlation (Sec-
tion 1.3.1). This is particularly true for privacy policies as there are no standards for their
writing. Certain privacy policies frequently omit sections related to certain topics, e.g., if
a topic does not concern the website in question. Recently, Cranor et al. (Cranor et al.,
2015) showed through their analysis of 75 online policies that many policies do not pro-
vide enough transparency about data collection for the users to make informed privacy
decisions. Using a larger data set ensures that keywords that are extracted cover as many
topics as possible.

In this chapter, results for the performance of various state-of-the-art unsupervised key-
word and keyphrase extraction algorithms over the online privacy policy domain are pre-
sented. By employing different kinds of algorithms, the hope was to find an algorithm that
was best suited to tackle the unique features of the privacy policy text. In so doing, re-
duce the amount of time it takes to populate a taxonomy, and by extension construct an
ontology for the online privacy domain. As the domain of online privacy evolves and laws
change, we anticipate that these techniques will be transferable to the evolving privacy
policies. Moreover, automating keyword extraction should reduce the cost of ontology con-
struction, as fewer domain experts (lawyers and privacy researchers) will need to be hired
to manually do this task. The approach was divided into two parts:

i Compare human and algorithm performance for tackling the semantic diversity and
textual ambiguity of 21 privacy policies.

ii To reduce the problem of topic relation and to reinforce the results found in the first
experiment, investigate how the algorithms perform over a large and diverse set of 631
privacy policies. This ensures that the results are representative of the problem do-
main.
2.1 Experiment I: Initial Benchmarking

For this experiment, 21 online privacy policies from a variety of domains were selected. To ensure diversity, policies were selected based on the their:

i Length
ii Transparency
iii Comprehension (level of difficulty)
iv Intended geographic audience (U.S., E.U, Canada)
v Industry sectors (healthcare, e-commerce, etc.)
vi Most visited websites

Table 2.1 shows the breakdown of the domain specific corpus. These policies were a subset of the larger data set used in Experiment 2.2. For keyword extraction the following methods were applied: manual, TF-IDF, RAKE, TextRank, and AlchemyAPI. An overview of Experiment 1 is shown in Figure 2.1.

1As listed under: https://en.wikipedia.org/wiki/List_of_most_popular_websites
Table 2.1: Breakdown of the 21 privacy policy corpus for Experiment I.

<table>
<thead>
<tr>
<th>Domain</th>
<th>No. of websites selected</th>
</tr>
</thead>
<tbody>
<tr>
<td>Healthcare</td>
<td>1</td>
</tr>
<tr>
<td>Insurance</td>
<td>2</td>
</tr>
<tr>
<td>Banking &amp; Financial</td>
<td>5</td>
</tr>
<tr>
<td>E-Commerce</td>
<td>3</td>
</tr>
<tr>
<td>File Sharing</td>
<td>1</td>
</tr>
<tr>
<td>Search Engines</td>
<td>2</td>
</tr>
<tr>
<td>Social Networking</td>
<td>3</td>
</tr>
<tr>
<td>EU Specific</td>
<td>3</td>
</tr>
<tr>
<td>Cloud Hosting</td>
<td>1</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>21</strong></td>
</tr>
</tbody>
</table>

2.1.1 Five Key Terms Extraction Methods

In our research, the four most widely used unsupervised algorithms for keyword extraction were chosen for investigation; along with the more recently introduced AlchemyAPI from IBM. This section briefly describes how they work along with our methodology for manually selecting key terms. A brief overview of this process is given in Figure 2.2.

Manual Extraction

As no previous attempt has been undertaken to create a generic taxonomy for online privacy policies, there exists no baseline for comparison. In order to compare the algorithms, terms were manually extracted from 21 privacy policies. The terms were carefully selected to create a comprehensive taxonomy that could be used to create a general ontology over the online privacy domain. The criteria used for term selection are described in Table 2.2.

Figure 2.2: Creating a baseline for the evaluation of keyword extraction algorithms.
Table 2.2: Criteria for manually extracting key terms.

<table>
<thead>
<tr>
<th>Concept</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Legal terms</td>
<td><em>Online Privacy Protection Act, non-disclosure agreement</em></td>
</tr>
<tr>
<td>Legal organizations (government, regulatory, commercial, and computing organizations)</td>
<td><em>federal trade commission</em></td>
</tr>
<tr>
<td>Acronyms of legal organizations and acts</td>
<td><em>FTC, COPPA</em></td>
</tr>
<tr>
<td>Legal entities that can be used to define an organization or an individual</td>
<td><em>personal information, address, account id, internet protocol address</em></td>
</tr>
<tr>
<td>Data sharing</td>
<td><em>3rd party cookies, aggregate information, google analytics</em></td>
</tr>
<tr>
<td>Hosting</td>
<td><em>backup storage, servers</em></td>
</tr>
<tr>
<td>Web &amp; tech related terms</td>
<td><em>ad data, cookies, analytics, tracking cookies</em></td>
</tr>
<tr>
<td>Legal actions and legal processes</td>
<td><em>tracking, surveillance</em></td>
</tr>
<tr>
<td>Mobile privacy</td>
<td><em>geo-location, device identification</em></td>
</tr>
</tbody>
</table>

**TF-IDF**

TF-IDF ([Sparck Jones, 1972](#)) is a commonly used baseline when comparing keyword extraction algorithms. For a given phrase \((j)\) in a text document \((k)\) from a corpus of size \(n\), its term weight \((w_{jk})\) is the product of its term frequency \((tf_{jk})\) and the inverse document frequency \((idf_j)\).

\[
w_{jk} = tf_{jk} \times idf_j \tag{2.1}
\]

\[
idf_j = \log_2 \left( \frac{n}{df_j} \right) \tag{2.2}
\]

Essentially, it describes the importance of a term to a document whilst still taking into consideration its occurrence in the corpus.

**RAKE**

RAKE ([Rose et al., 2010](#)) is an unsupervised, domain-independent, language-independent, and corpus-independent method for extracting keywords. It is based on the observation that keywords frequently contain multiple words but rarely contain punctuation marks or stop words. Stop words include function words such as *and*, *the*, and *of*, or other words with little lexical meaning. RAKE takes a list of stop words, a list of keyphrase delimiters, and a set of word delimiters as input, and uses them to partition the document into candidate keywords. Co-occurrences of words within these candidate keywords are meaningful and allow us identify word co-occurrence without the need for arbitrarily sized sliding windows. This way, word associations are captured in a way that adapts to text provided
resulting in adaptive and fine-grained word co-occurrences that will be used to score candidate keywords. The score of each candidate keyword is the sum of scores of each of its member term co-occurrences. Word scores are based on two things: (i) its word frequency \( \text{freq}(w) \), its degree \( \text{deg}(w) \) (within the co-occurrence matrix), and the ratio of degree to frequency \( \frac{\text{deg}(w)}{\text{freq}(w)} \).

RAKE is also able to find key terms with stop words within them, e.g. ‘axis of evil’, by finding pairs of candidate keywords that are adjoining and in the same order in a document. The stop word is then filled in and the score is calculated as usual. RAKE evaluates each document independently. Since, there is no standard language used by privacy policies, it was expected that this feature would be useful in finding relevant terms.

**TextRank**

TextRank [Mihalcea & Tarau, 2004] is a well-known graph-based keyphrase extraction algorithm. It uses graph based ranking model for graphs extracted from texts to rank keywords based on the co-occurrence matrix between words. Formally, let \( G = (V, E) \) be a directed graph of a set of vertices \( V \) connected by a set of edges \( E \), where \( E \) is a subset of \( V \times V \). Each vertex corresponds to a word type. A weight \( w_{ij} \) is assigned to the edge connecting two vertices, \( V_i \) and \( V_j \), and its value is the number of times the corresponding word types co-occur within a window of \( W \) words in the associated text. The score of a vertex \( V_i \) is defined by equation 2.3 (Brin & Page, 2012):\n
\[
S(V_i) = (1 - d) + d \times \sum_{V_j \in \text{In}(V_i)} \frac{w_{ji}}{\sum_{V_k \in \text{Out}(V_j)} w_{jk}} S(V_j)
\]

(2.3)

Where \( \text{In}(V_i) \) is a set of vertices that point to a vertex \( V_i \) (predecessor), \( \text{Out}(V_i) \) is a set of vertices that the vertex \( V_i \) points to, and \( d \) is the damping factor that can be set between 0 and 1 (usually set 0.85). Scores for the vertices determine their importance, and the word types that correspond to the highest scored vertices form keyphrases. The score for \( V_i, S(V_i) \), is initialized with a default value and is computed in an iterative manner until convergence.

**AlchemyAPI**

AlchemyAPI was chosen as a state-of-art industrial tool for key term extraction. Its implementation is available via an open API. It takes as input a text document and returns a list of significant words or phrases extracted from the document. Since it returns a set of ngram terms, it was still possible to compare the results with the other algorithms. Due to the proprietary nature of the algorithm, no information could be found about its implementation.

[^2]: [http://www.alchemyapi.com/products/alchemylanguage/keyword-extraction](http://www.alchemyapi.com/products/alchemylanguage/keyword-extraction)
2.1.2 Experimental Setup

In this section, the parameters used for the algorithms, pre-processing, and evaluation methods that were used in the extraction of the keywords are described.

Pre-processing

To ensure consistency of key term extraction across the corpus, a series of pre-processing steps were employed to normalize the input text:

i All of the input was first converted to lowercase.

ii URLs and email addresses were removed.

iii Non-printable characters (as defined by the `string.printable` set in Python3) were removed.

iv Other special characters that were not caught by previous filters (*@#), as well as other ASCII based characters from the `string.punctuation` set in Python3 were removed.

v Tokenized numbers were also removed as they do not tend to add value to the taxonomy e.g. ‘1945’.

The standard Porter Stemming Algorithm\cite{Porter1980} was used from the NLTK\footnote{http://www.nltk.org/} library to consolidate inflected word forms to their root, e.g. ‘collection’, ‘collecting’, ‘collected’ all refer to the same concept of ‘collect’. This was done to reduce the number of variations of a term so as to not skew the validation. Furthermore, the duplicates were removed from the resulting sets.

Algorithm Setup

This section clarifies the configurable parameters of the algorithms that were used in the experiments.

**TF-IDF:** Scikit-learn’s `TfidfVectorizer`\footnote{http://scikit-learn.org/stable/} was used to implement this algorithm. TF-IDF typically operates on single words, but the Scikit tool allows `ngrams` to be extracted. To encourage the extraction of multi-word terms, the `ngram` parameter was set to (1,3) to allow the extraction of terms of length 1-3. To further improve the relevancy of candidate terms, only the top 1/5th of the scoring terms per document were selected.

**RAKE:** A well-known Python library\footnote{https://github.com/aneesha/RAKE} of this algorithm was used in implementation for the experiment.
TextRank For Part-of-speech (POS) tagging, NLTK’s recommended Maxent Treebank Tagger trained on the PennParsed Corpora was used. The syntactic filters that achieved the best results recommended in the original TextRank proposal were used: adjectives and nouns. This was also done after consulting the results of the manual annotation. Other parameters were also chosen exactly like those that performed best in the TextRank experiments: the co-occurrence window of 2 for term relationships and undirected treatment of edges between vertices.

Evaluation Metrics

As it is traditionally done with information retrieval, the results from each of the algorithms were compared using 3 main scores (Section 1.4.3): precision, recall, and f-measure. To further understand the similarities in between the resultant sets Jaccard’s similarity co-efficient was computed.

2.1.3 Results

The key terms generated by the algorithms were compared with the manually extracted set. In order to find the best parameters to improve the scores for TF-IDF, the document frequency (df) score for each term was combined with the TF-IDF scores as demonstrated by Jacques Vergne (Vergne, 2004); results are shown in Table 2.3. Due to this, the high $F_1$-score of 27% was achieved by keeping the minimum document frequency to 3. The parameters used for the algorithms that achieved the best $F_1$-score in Experiment I (Section 2.1) were also used in Experiment II (Section 2.2).

The results for Experiment I are summarized in Table 2.4. The manual annotation resulted in 829 terms being recognized as important for building a taxonomy for the online privacy domain. Results show no difference between TextRank and AlchemyAPI’s performance. TF-IDF performed the best with RAKE was following a close second.
Table 2.4: Results for Experiment I: Applying keyword extraction algorithms to a small corpus of online privacy policies.

<table>
<thead>
<tr>
<th></th>
<th>TF-IDF</th>
<th>RAKE</th>
<th>TextRank</th>
<th>AlchemyAPI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Terms</td>
<td>907</td>
<td>4116</td>
<td>864</td>
<td>799</td>
</tr>
<tr>
<td>Precision</td>
<td>25%</td>
<td>13%</td>
<td>17%</td>
<td>18%</td>
</tr>
<tr>
<td>Recall</td>
<td>28%</td>
<td>66%</td>
<td>18%</td>
<td>17%</td>
</tr>
<tr>
<td>F1-score</td>
<td>27%</td>
<td>22%</td>
<td>18%</td>
<td>18%</td>
</tr>
<tr>
<td>JSC</td>
<td>0.85</td>
<td>0.88</td>
<td>0.90</td>
<td>0.90</td>
</tr>
</tbody>
</table>

Table 2.5: Breakdown of the corpus of privacy policies for Experiment II.

<table>
<thead>
<tr>
<th>Domain</th>
<th>No. of websites selected</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adult</td>
<td>12</td>
</tr>
<tr>
<td>Business</td>
<td>41</td>
</tr>
<tr>
<td>Children</td>
<td>38</td>
</tr>
<tr>
<td>Gaming</td>
<td>67</td>
</tr>
<tr>
<td>Healthcare</td>
<td>32</td>
</tr>
<tr>
<td>Recreation</td>
<td>32</td>
</tr>
<tr>
<td>Reference</td>
<td>52</td>
</tr>
<tr>
<td>Science</td>
<td>66</td>
</tr>
<tr>
<td>Shopping</td>
<td>179</td>
</tr>
<tr>
<td>Sports</td>
<td>47</td>
</tr>
<tr>
<td>Technology</td>
<td>67</td>
</tr>
<tr>
<td>Total</td>
<td>631</td>
</tr>
</tbody>
</table>

2.2 Experiment II: Enlarged Dataset

In Experiment II, algorithms were applied against 631 privacy policies as developed by the Data Management and Privacy Governance Lab at the University of Guelph. The algorithm with the highest $F_1$-score was used as a baseline for evaluating the rest. This baseline, as created by the AlchemyAPI, was chosen by comparing the results of all the algorithms against the manual annotation from the first experiment. An overview of Experiment II is shown in Figure 2.3.

2.2.1 Corpus

As with Experiment I, described in Section 2.1, the selection criteria for privacy policies was the same. A detailed domain breakdown of the policies is shown in Table 2.5.
Figure 2.3: Experiment II: Running keyword extraction algorithms over a corpus of online privacy policies.
Table 2.6: Results for Experiment II: Applying keyword extraction algorithms to a corpus of online privacy policies.

<table>
<thead>
<tr>
<th></th>
<th>AlchemyAPI (vs. Manual)</th>
<th>TF-IDF</th>
<th>RAKE</th>
<th>TextRank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Terms</td>
<td>12694</td>
<td>52736</td>
<td>47777</td>
<td>6951</td>
</tr>
<tr>
<td>Precision</td>
<td>3%</td>
<td>6%</td>
<td>16%</td>
<td>11%</td>
</tr>
<tr>
<td>Recall</td>
<td>40%</td>
<td>24%</td>
<td>60%</td>
<td>6%</td>
</tr>
<tr>
<td>F$_1$-score</td>
<td>5%</td>
<td>9%</td>
<td>25%</td>
<td>7%</td>
</tr>
<tr>
<td>JSC</td>
<td>0.98</td>
<td>0.95</td>
<td>0.85</td>
<td>0.96</td>
</tr>
</tbody>
</table>

2.2.2 Results

The experimental setup of this experiment was same as the first experiment for all four algorithms. Instead of choosing TF-IDF as the baseline as is generally done, the results from all of the algorithms were evaluated against the results from the manual annotation. AlchemyAPI had the highest F$_1$-score of 4.85%; as such, it was chosen as the baseline to evaluate the results from the rest of the algorithms. The results are summarized in Table 2.6.

2.3 Discussion

Results from Experiment I (Section 2.1) show no significant difference between TextRank and AlchemyAPI’s performance when compared to manual extraction. TF-IDF performed the best with an F$_1$-score of 27% and RAKE was a close second with 22%. However, in Experiment II (Section 2.2) using the larger data set, AlchemyAPI performed the best with RAKE a close second.

From Table 2.3 it is clear that the most important keywords appeared at least thrice in the same policy. It was unexpected however for TF-IDF to perform as well as it did, since the language used in all of the policies were quite different; some were legalese (healthcare, financial institutions), and others easier to read (Dropbox, Google). This diversity is important because TF-IDF scores each term not only by how it appears in one document but also its scarcity of appearance elsewhere in the corpus. As the language across the corpus was inconsistent, it was expected that TF-IDF would perform the poorest.

On the other hand, RAKE, TextRank, and AlchemyAPI only consider a single document at a time when extracting terms instead of the entire corpus. For these algorithms, a per document analysis should have yielded better results and higher scores because certain topics are excluded from certain privacy policies if they do not concern websites. For example, not all privacy policies have information on cookies; cookie related keywords would be absent in these documents. However, in Experiment I (Section 2.1) these algorithms did not perform as well.
For this reason, it was expected that evaluating a larger data set of privacy policies would eliminate, or at least reduce, the gaps in the individual algorithm performances; it did not. One reason for this could be that since the AlchemyAPI algorithm was used as a benchmark, it could be negatively affecting the scores in the second experiment as calculating scores based on it could bias the results in favour of algorithms that evaluate each document separately. In contrast, keyword extraction done by domain experts would likely be more neutral. For the future, we anticipate that increased manual annotation will provide a better baseline to score the algorithms against. It might also be useful to re-evaluate the results against another keyword extraction algorithm.

Our results may have been affected by four major types of errors: over-generation, redundancy, infrequency, and evaluation errors as described by Hasan and Ng (Hasan & Ng, 2014).

**Over-generation errors** are a type of precision errors that occur when the algorithm correctly chooses a candidate term because it contains a word that appears frequently in the document or corpus, but at the same time outputs similar but incorrect candidate terms because they too contain a frequently occurring word. This contributes to a high degree of divergence within the results for that algorithm. A prime examples of this are the results from the TF-IDF which performed very poorly because it reported *unigrams*, *bigrams* and *trigrams* that were often subsets of each other resulting in a very high number of terms being generated. RAKE and TextRank did not suffer much from this problem because both calculate scores for candidate keyphrases based on frequency of groups of constituent words appearing in that particular order whereas TF-IDF treats each of its constituent word as a separate word, in order to calculate its score, and sums individual word scores to calculate the candidate keyphrase score. An example of this is illustrated in Fig 2.4.

**Redundancy errors** are a type of precision error that occur when an algorithm correctly identifies a candidate term or phrase, whilst simultaneously selecting a similar candidate term or phrase that are semantically the same. For example, ‘account use’ and ‘account usage’ refer to the same concept of anonymous usage, but are reported as separate keywords. All the algorithms were affected by this error. However, TextRank had fewer errors overall because of the hard limit of the co-occurrence window to 2 words per keyphrase.

**Infrequency errors** are a type of recall error that occur when an algorithm does not
recognize a correct candidate keyphrase because of its scarcity in a document. It is challenging to address this type of error because algorithms rarely choose candidate terms that only appear once or twice in a document. Furthermore, for TF-IDF all candidate terms were forced to appear at least thrice in a document to generate more relevant results.

**Evaluation errors** are a type of recall error that occurs when the scoring algorithm incorrectly scores a keyphrase as not a keyword despite it being syntactically similar to a gold keyphrase. Part of this type of error was dealt with by scoring the stemmed output of the keyphrases. Still there was not a concrete way of eliminating all of the errors. For example, ‘account use’ and ‘account usage’ should be recognized as the same concept but is calculated as separate keyphrases.

It is important to remember that these results are preliminary and only reflect results of 4 well known algorithms. The experiments were primarily conducted to see how effective keyword extraction algorithms were over a semi-legal and legal domain such as online privacy. Further research will entail running more algorithms over a larger corpus. It is expected that increasing the number of policies will increase the diversity of the privacy policies by enough that terms currently viewed as unimportant due to the low occurrence in the corpus might be scored higher for TF-IDF. Topic-based clustering such as CommunityClustering (Grineva, Grinev, & Lizorkin, 2009) may also prove promising for future research.

### 2.4 Conclusion

In this chapter, a systematic study of keyword and keyphrase extraction over the domain of online privacy policies was presented. Four prominent automatic keyword extraction algorithms were tested against manual annotation and it was found that TF-IDF performs well against manual annotations on a small data set of 21 policies, and AlchemyAPI performs marginally better than TF-IDF on larger data sets.

Due to the lack of standardized language used in online privacy policies, algorithms that evaluated individual documents (TextRank, RAKE, AlchemyAPI) outperformed by a significant margin algorithms that extracted keywords from the larger corpus (TF-IDF). Our results confirm that using NLP techniques for keyword and keyphrase retrieval from legal and semi-legal documents remains a challenging task.

Our preliminary results will guide further research in the field of online privacy policies and machine learning. For these experiments the entire data set of heterogenous privacy policies were used; it would be worthwhile to evaluate the performance of algorithms against data sets for specific privacy policy domains e.g. cloud service providers, e-commerce, and social networking. Similarly, algorithm performance could be assessed for policies that are divided into common sections like share, collect, consent, and store.

To compare the performance of supervised algorithms for keyword extraction against the unsupervised algorithms tested in this chapter, in Chapter 3, a supervised learning algorithm was implemented: KEA (Witten et al., 1999) is a content-based single document
algorithm which used our manually extracted terms as a training set. In addition, the performance of trained keyword extractors was tested against the algorithms and against domain experts (privacy researchers and lawyers).
Keyword/keyphrases extraction remains a difficult task; the state-of-the-art performance of keyword extraction algorithms hovers around 20-30% (Kim, Medelyan, Kan, & Baldwin, 2010). Keyword (or keyphrase) extraction has been historically used to recognize key topics and concepts in documents. This task involves identifying and ranking candidate keywords based on the relatedness to the document. Keyword extraction algorithms utilize various techniques to perform their task: statistical learning, part-of-speech (POS) tagging, lexical and syntactic feature extraction. Generally, they work in two steps:

1. Identifying candidate keywords/keyphrases from the document using heuristics.
2. Recognizing if the chosen candidate keywords/keyphrases are correct or not using supervised and unsupervised methods.

In Chapter 2, five unsupervised learning algorithms were used to extract keywords for the purpose of identifying key concepts with the goal of generating a taxonomy for the online privacy policy domain. The research was conducted in two experiments. In the first, the algorithms were evaluated over a smaller corpus where a set of manually extracted terms by the researcher was used as the baseline. Second, the algorithms were evaluated over a larger corpus where the results from the best performing algorithm from the first experiment was held as the baseline. While the algorithm TF-IDF achieved an F-measure of 27% over a small corpus (21 policies), over a large corpora (631 policies) algorithms evaluating single documents individually, such as AlchemyAPI and TextRank, performed the best.

As previously mentioned, our results may have been affected by the following errors: over-generation, redundancy, infrequency, and evaluation. Since the unsupervised algorithms focus on the task of ranking and/or clustering based on semantic and lexical analysis, these errors are a result of language used in the privacy policies which tends to be inconsistent
The alternative to unsupervised learning algorithms are supervised learning algorithms in which a model is first trained on a set of manually extracted terms, and thus the task becomes one of classification, i.e., whether a candidate keyword should be classified as a document keyword or not. In case of privacy policies, it was expected that the results would improve because training a model should reduce various errors that occurred with unsupervised learning. For example, since the keywords are first being extracted manually; if the terms are infrequent, the trained model would learn this bias reducing infrequency errors.

### 3.1 Study Design

The primary objective of this experiment was to test whether a supervised learning algorithm could outperform the unsupervised learning algorithms used in our previous study. As such, choosing an effective supervised challenger was key. To compare unsupervised algorithms, a supervised learning algorithm, KEA, was investigated. KEA is an effective supervised learning algorithm utilizing a Naive Bayes algorithm for training learning models (Witten et al., 1999).

It works in two phases: training and model creation, and extraction. In the training phase, candidate keyphrases are selected both from the training documents and the corpus, features (attributes) are calculated, and keyphrases determined. Candidate keyphrase selection works in three phases: text pre-processing, identifying the candidate keyphrases, and stemming and case-folding. Features that are calculated are, TF-IDF and first occurrence (the first time the term occurs in a document), which are then discretized for the machine learning scheme. Finally, the keyphrase are determined from the discretized values using the Naive Bayes technique (Lewis, 1998). In the extraction phase, the candidate keyphrase selection is repeated on the documents to calculate feature values. Then the Naive Bayes algorithm is used along with the values calculated in the model to determine if a candidate keyphrase is a keyphrase or not.

The study was broadly broken up into two parts: Experiment I examines manual extraction by trained non-domain experts, while Experiment II compares supervised and unsupervised techniques.

### 3.2 Experiment I: Manual Keyword Extraction

An overview of Experiment I is shown in Figure [3.1](#).
3.2.1 Participants

Four participants were selected for this experiment. Since this was a preliminary study conducted to test if non-experts can be trained enough to extract important keywords, we thought that 4 participants was enough. All had a graduate level education in Computer Science with varying knowledge of online privacy and were male with a mean age of 33.65 (range 22 – 60). None had a research background in privacy and they rarely read website privacy policies.

3.2.2 Procedure

The participants were briefed on intelligent reasoning systems and taxonomies to ensure they understood the basic concepts and how their keywords would be used. The same set of criteria for manual extraction, as in the previous chapter (Table 2.2), was provided to help the participants select the appropriate keywords and keyphrases. In order to have some time limit, it was estimated that it took less than 2 hours to read and annotate 5 policies. Hence, the participants were given 2 hours to read 5 privacy policies (a subset of the 21 privacy policies used for the training model; see Table 2.1) and highlight terms (unigrams, bigrams, trigrams, etc.) they thought were important: concepts, themes, and terms; pertaining to the online privacy domain and as outlined in the criterion. The 5 policies selected were from different industry sectors; intended for a diverse audience; and conformed to the laws of multiple countries; they included: Google, Facebook, UEFA,
Royal Bank of Canada, and Wal-Mart (including policy for California). Over a 2 hour period, each participant was presented with the privacy policies in a different order to reduce the possibility of an ordering bias.

Participants used an open source program called ‘Skim’ for annotation. A Python script was then used to extract all of the highlighted keywords and store the results in a comma separated values (CSV) file for further analysis.

To ensure consistency of key term extraction across the data sets, the following post-processing steps were taken to normalize the text:

1. All of the terms were first converted to lowercase.
2. Non-printable characters (as defined by the `string.printable` set in Python3) were removed; and the remaining special characters that were not caught by previous filters (*@#), as well as other ASCII based characters from the `string.punctuation` set in Python3 were removed.
3. Tokenized numbers were also removed as they do not tend to add value to the taxonomy e.g. ‘1945’.
4. The standard Porter Stemming Algorithm (Porter, 1980) was used from the NLTK library to consolidate inflected word forms to their root.
5. Finally, duplicates were removed from the resulting sets.

### 3.2.3 Results

First, the data collected from all of the participants were compared to the data set generated by the primary researcher. The results are shown in Table 3.1. It must be noted that participant 3 only completed 3 of the 5 policies because he found reading some policies quite challenging and hence taking longer to read. He also reported to initially having trouble understanding the task. In general, participants reported that policies were repetitive and often vaguely described their intent with regard to collecting personal information. When asked to state which privacy policy was most clear and readable, Facebook was described as the most transparent with UEFA being the least. The highest $F_1$-score was 59% with a mean of 51.75%.

In order to test the collective efficacy of the annotations, the researcher’s data set was compared with the combined data set of all of the participants. The results are reported in Table 3.2.

Finally, all five data sets were compared to each other by holding one of the data set as the baseline and comparing it with the rest. The results are reported in Table 3.3. The mean of all of the values is 52.1%, which agrees with the previous analysis in Table 3.2.

---

Table 3.1: Results from manual keyword extraction by participants.

<table>
<thead>
<tr>
<th>Terms</th>
<th>Researcher 1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Terms</td>
<td>560</td>
<td>581</td>
<td>650</td>
<td>353</td>
</tr>
<tr>
<td>Precision</td>
<td>49%</td>
<td>55%</td>
<td>59%</td>
<td>56%</td>
</tr>
<tr>
<td>Recall</td>
<td>51%</td>
<td>64%</td>
<td>37%</td>
<td>51%</td>
</tr>
<tr>
<td>F1-score</td>
<td>50%</td>
<td>59%</td>
<td>45%</td>
<td>53%</td>
</tr>
<tr>
<td>JSC</td>
<td>0.67</td>
<td>0.58</td>
<td>0.71</td>
<td>0.64</td>
</tr>
</tbody>
</table>

Table 3.2: Comparing performance of manual extraction primary researcher vs. combined data set generated by participants.

<table>
<thead>
<tr>
<th></th>
<th>Researcher</th>
<th>Participants</th>
</tr>
</thead>
<tbody>
<tr>
<td>Terms</td>
<td>560</td>
<td>1038</td>
</tr>
<tr>
<td>Precision</td>
<td>40%</td>
<td></td>
</tr>
<tr>
<td>Recall</td>
<td>75%</td>
<td></td>
</tr>
<tr>
<td>F1-score</td>
<td>52%</td>
<td></td>
</tr>
<tr>
<td>JSC</td>
<td>0.65</td>
<td></td>
</tr>
</tbody>
</table>

This is significantly higher than the 20 – 30% performance of most state-of-the-art keyword extraction algorithms.

3.3 Experiment II: Supervised Learning

For this experiment, a corpus of 631 privacy policies as developed by the Data Management and Privacy Governance Lab at the University of Guelph was used to evaluate the supervised learning algorithm- KEA. This is the same corpus that was used to evaluate unsupervised learning algorithms in the previous Chapter 2.

Table 3.3: Comparing $F_1$-scores between participants’ and researcher’s data sets.

<table>
<thead>
<tr>
<th></th>
<th>Researcher</th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Researcher</td>
<td>-</td>
<td>50%</td>
<td>59%</td>
<td>45%</td>
<td>53%</td>
</tr>
<tr>
<td>P1</td>
<td>50%</td>
<td>-</td>
<td>56%</td>
<td>46%</td>
<td>52%</td>
</tr>
<tr>
<td>P2</td>
<td>59%</td>
<td>56%</td>
<td>-</td>
<td>50%</td>
<td>58%</td>
</tr>
<tr>
<td>P3</td>
<td>45%</td>
<td>46%</td>
<td>50%</td>
<td>-</td>
<td>55%</td>
</tr>
<tr>
<td>P4</td>
<td>53%</td>
<td>52%</td>
<td>58%</td>
<td>55%</td>
<td>-</td>
</tr>
</tbody>
</table>
3.3.1 Part I

In the first part of the experiment, a set of 21 policies were used for manual extraction of keywords which in turn were used to train the learning model for KEA. These are the same 21 policies that were used in the previous chapter for unsupervised learning. A breakdown of these policies is summarized in Table 3.1. An overview of Experiment II Part I is shown in Figure 3.2.

Results

Once a model was trained over the manually extracted set of keywords, KEA was then run over the entire corpus. Results from the algorithm were then compared with the results from unsupervised algorithms (see Table 3.4). Initial results show that the supervised algorithm performed better than the unsupervised ones but not significantly.
Table 3.5: Comparing $F_1$-scores between participants’ and researcher’s generated data sets from KEA.

<table>
<thead>
<tr>
<th>Researcher</th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Researcher</td>
<td>-</td>
<td>62%</td>
<td>70%</td>
<td>64%</td>
</tr>
<tr>
<td>P1</td>
<td>62%</td>
<td>-</td>
<td>71%</td>
<td>75%</td>
</tr>
<tr>
<td>P2</td>
<td>70%</td>
<td>71%</td>
<td>-</td>
<td>67%</td>
</tr>
<tr>
<td>P3</td>
<td>64%</td>
<td>75%</td>
<td>67%</td>
<td>-</td>
</tr>
<tr>
<td>P4</td>
<td>72%</td>
<td>72%</td>
<td>79%</td>
<td>67%</td>
</tr>
</tbody>
</table>

Table 3.6: Comparing performance of keyword extraction from KEA based on primary researcher vs. combined data set generated by participants.

<table>
<thead>
<tr>
<th>Researcher</th>
<th>Participants</th>
</tr>
</thead>
<tbody>
<tr>
<td>Terms</td>
<td>30261</td>
</tr>
<tr>
<td>Precision</td>
<td>57%</td>
</tr>
<tr>
<td>Recall</td>
<td>93%</td>
</tr>
<tr>
<td>$F_1$-score</td>
<td>71%</td>
</tr>
<tr>
<td>JSC</td>
<td>0.45</td>
</tr>
</tbody>
</table>

3.3.2 Part II

To demonstrate that KEA performs well over smaller training sets, and better when results across multiple annotators are combined, in the second part of the experiment, the model was trained on the annotation for only the 5 policies chosen for Experiment I (Section 3.2). The trained model was then tasked with extracting keywords and keyphrases from the rest of the corpora (626 policies). This was done for all the participants and the results were compared against each other; reported in Table 3.5. An overview of of Experiment II Part II is shown in Figure 3.3.

Since the training set only contained 5 privacy policies with roughly half of the terms present amongst all of the participants, the training set did not contain enough variance to train distinctly different models. The obvious exception being participant 3 who only annotated 3 policies.

Once again, participants’ data sets were combined and compared against the researcher’s data set of generated keywords to see if the accuracy rose with more number of annotators; the results are presented in Table 3.6. The combined data set achieves a score of $F_1$-score of 71%. One reason why the scores are different for the combined data sets (Table 3.2) is due to the smaller training set for the third participant. With less labelled data to train with, the training algorithm could overgeneralize what is not considered a keyword and hence ignore a large number of candidate keywords.
Figure 3.3: Experiment II Part II: Evaluating KEA results trained on non-domain experts’ data set of manually extracted keywords from privacy policies.
3.4 Discussion

The mean $F_1$-score of $52\%$ in Experiment I (Table 3.1) demonstrates an important aspect of keyword extraction with privacy policies—keyword extraction with privacy policies is a hard task. While $52\%$ is a score better than the $20 - 30\%$ performance of most state-of-the-art keyword extraction algorithms, a qualitative analysis of the annotated keyword-s/keywords/keyphrases by the participants suggests that despite having a concrete set of rules, examples, and training, participant’s understanding of the technical terms and the text can still result in a diverse and non-overlapping set of terms. What one participant considers important concept not shared by a peer. This was mostly true for the more ambiguous and less technical parts of the policies, while technical details were easily picked up by all participants. Since all of the participants had a background in Computer Science, technical details would be easy to comprehend and more transparent to them. However, this might not be the case if the participant had no technical background; then, everything would have been equally less transparent. When written ambiguously privacy policies are difficult to comprehend.

In Part I of Experiment II it was found that the supervised learning algorithm improved the $F_1$-score of keywords being extracted. This is important for two primary reasons: quality and cost. One of the most time consuming tasks involved in generating a taxonomy is capturing major important themes and concepts of the target domain. This is where keyword extraction plays an important role in reducing the time taken to capture all of the themes and concepts. In this case, instead of reading a large number of privacy policies to identify all of the important themes and concepts, supervised learning promises to be a viable alternative. The generated keywords act as candidate terms that can be used to enrich a taxonomy, thus reducing the cost and time of reading a large number of privacy policies as well as improving the quality of the taxonomy by including terms that might have been covered in text that might not have been read due to resource restrictions.

Furthermore, the diversity of keywords/keyphrases found, in Experiment I, between participants can be used to improve the training data for supervised learning. The training data in Experiment II Part I was generated by a single researcher, it might be useful for another researcher or domain expert to read a set of non-overlapping policies and generate another set of training data. This would not only validate the current training data but also create more labelled data to train the model on. It would also prove helpful to review the candidate terms generated by participants in the first experiment and enrich the present training data, i.e., carefully merge all of the data sets to create a more comprehensive data set that captures all relevant keywords/keyphrases including the ones that might have been missed by individuals within the group.

Part II of the second experiment showed that if non-domain experts are given sufficient training, a supervised learning algorithm trained with their labelled data set could result in a training model that is able to extract most of the keywords and keyphrases that are being extracted with the help of training model, and built with labelled data from domain experts. Hence, it is possible to reduce cost and train non-experts and extract keywords and keyphrases with reasonable success.
Currently, privacy policies are heterogeneous as there are no laws/guidelines that mandate a certain structure, concepts, or terminology. This makes finding, identifying and understanding relevant information a time consuming task. By utilizing an intelligent reasoning system and mapping important concepts, ideas, and themes our work helps to identify important sections in an unstructured privacy policy; thus, resulting in less time needed to find important information in policies improving transparency and making polices more usable. It could further be used to introduce structure in future policies.

### 3.5 Conclusion

In this chapter, our previous work on keyword and keyphrase extraction over the domain of online privacy policies was extended. In our first experiment, the difficulty of extracting keywords from privacy policies was demonstrated and the challenges associated with this task was discussed. In Experiment II, a supervised algorithm for keyword extraction was applied, and its superior performance over unsupervised algorithms applied to the same corpus of 631 online privacy policies was demonstrated. Our results confirmed that using natural language processing techniques for keyword and keyphrase retrieval from privacy policies remains a challenging task.

Our preliminary results will guide further research in the field of online privacy and machine learning, and making policies more transparent and usable. We intend to improve our training set by having other domain experts (researchers and lawyers) identify key concepts, ideas, and themes in a non-overlapping set of privacy policies. In addition, it will be useful to compare how trained supervised algorithms perform against domain experts. In the following chapter, our ontology for the online privacy domain is introduced using the keywords and keyphrases extracted from the the experiments within this and previous chapters.
Chapter 4

A Domain Ontology For Online Privacy: DOOP

An ontology provides a logical framework that can be used to construct models which facilitate a pre-determined purpose, e.g., communication, agent, and domain models. These models are in turn used for processing information (Mommers, 2001). A domain ontology is built using a domain targeted corpus and captures the vocabulary and conceptualizes the domain. From a semantic standpoint, a memorandum of understanding (MOU) document which has legal implications such as an online privacy policy contains six features: rules, norms, legal institutions, legal acts, technical terms, and the relations between such entities (Mommers, 2001; Biagioli, Francesconi, Passerini, Montemagni, & Soria, 2005). Relationships between these features can be captured in a structural overview like a legal ontology. As previously discussed in Section 1.2, there are two basic parts to an ontology: taxonomy and relationships. Chapters 2 and 3 described how unsupervised and supervised keyword extraction algorithms were used to extract keywords of the taxonomy of DOOP. This chapter introduces relationships between: the taxonomy, the privacy categories, and the queries; in addition to describing how the final ontology was constructed and validated. An overview of this chapter is shown in Figure 4.1 below.

Since the invention of ontologies in the early 90s, ontology engineering has remained more of an art form rather than an engineering process with rigid rules (Malheiros & Freitas, 2013). Which is to say, there is no one correct way of creating an ontology, rather the development differs depending on the ontology engineer and its purpose. However, several methodologies have been proposed to standardize the process of creating ontologies, e.g.: Methontology (Fernández-López, Gómez-Pérez, & Juristo, 1997), Onto-To-Knowledge (Staab, Studer, Schnurr, & Sure, 2001), Ontology 101 (Noy, McGuinness, et al., 2001), RapidOWL (Auer & Herre, 2007), and NeOn (Suárez-Figueroa, Gómez-Pérez, & Fernández-López, 2012), to name a few. A brief overview of these is presented in Section 4.1.

Despite their differences, they all share many commonalities. The two most important of these are iterative development and using competency questions (CQs) to define requirements. Some commonly used tools for ontology engineering include: Protégé-2000 (Noy et
Figure 4.1: Overview of how DOOP was constructed and validated.
Some of the more well-known ontology engineering methodologies are introduced (Section 4.1).

Our methodology for constructing the ontology is discussed (Section 4.2).

Our ontology is validated against CMU’s OPP-115 data set and details about the ontology are provided (Section 4.3).

Lastly, our findings and possible future areas of research are discussed (Section 4.4).

### 4.1 Ontology Engineering

This section provides a brief overview of popular methodologies used in ontology engineering.

#### 4.1.1 Methontology

Developed by the Ontological Engineering group at Universidad Politécnica de Madrid, this methodology enables the construction of ontologies at the knowledge level. It draws its inspiration from IEEE’s Standard for Developing Software Life Cycle Processes, which proposes a development process rooted in a life cycle based on changing prototypes. This methodology demonstrates techniques for carrying out the main activities identified by the IEEE Standard, i.e., specification, conceptualization, formalization, implementation, and maintenance. Ontology engineering tools, ODE and WebODE, support this methodology. This methodology can also be implemented via other tools.

#### 4.1.2 On-To-Knowledge

On-To-Knowledge was originally developed for standardizing the creation of enterprise ontologies with an emphasis on maintenance of Knowledge Management (KM) applications (applications that use ontologies at their core). It is based on iterative development

---

1 A system defined at the knowledge level is called the agent. It has three main components: goals, actions, and bodies. The agent’s behaviour is based on the principle of rationality, where knowledge is processed to determine which actions to take to achieve the agent’s goals. (Newell 1982)
and maintenance of ontologies through identification of goals that should be achieved by KM applications based on an analysis of the usage scenarios. It is composed of two broad steps, knowledge meta processes and knowledge processes. Knowledge meta process is defined as the process of identifying knowledge and introducing KM into an organization, and knowledge process is defined as the process of updating existing ontology and stressing knowledge creation. Knowledge meta process is divided into five main phases: feasibility study, kickoff, refinement, evaluation, and application and evolution; where the phases, refinement, evaluation, and application and evolution are run iteratively. Phases can further be divided into multiple steps. Knowledge process is divided into four main steps: knowledge creation and/or importation of documents and meta-data; capture of important knowledge from documents and meta-data via extraction and linkage; inferring additional knowledge through retrieval/access; and usage of knowledge in context. The On-To-Knowledge methodology’s name eponymous to the project first project that it was used on funded by the European Commission (Onderwijs 2017; Davies, Fensel, & Van Harmelen 2003).

4.1.3 Ontology 101

Ontology 101 provides a step-by-step methodology for creating simple ontologies iteratively using the Protégé-2000 (Musen 2015) ontology engineering tool, developed by Mark Musen’s research group at Stanford Medical Informatics. Since this methodology is geared towards the Protégé tool, it focuses more on declarative frame-based systems which are used to describe objects in a domain along with their properties rather than more complex and domain specific ontologies that can be constructed with the other methodologies mentioned in this section. This is intentional, because the methodology is intended for a novice audience, and is meant to be used as a learning tool to understand the capabilities of the Protégé tool. Overall, the authors emphasize the flexibility of ontology development and mention multiple design decisions that have to be made to fit the ontology’s purpose throughout the methodology.

4.1.4 RapidOWL

The proponents of RapidOWL recognize that older methodologies take a significant amount of capital investment, needed to define an ontology’s purpose and formulate CQs, similar to the waterfall model (Bell & Thayer 1976) from software engineering. This cost becomes progressively more expensive when large number of knowledge (ontology) engineers are involved as the ontology grows over time. To counter this, they propose a methodology based on the agile model (Beck et al. 2001), which advocates for evolutionary development and continuous improvement. This technique is significantly less costly when used with large development groups.
4.1.5 NeOn

In recognition of the large number of ontologies available online, the NeOn methodology promotes reusing and combining of ontologies to create new and networked ontologies drawing on multiple ontologies for their knowledge. NeOn provides a scenario-based framework to create ontologies and develop and expand ontology networks. The ontology networks play a vital role in a distributed and collaborative environments where ontologies are created by different domain experts and ontology engineers, who can be involved in various stages of an ontology’s development. Rather than a rigid work flow like Ontology 101, NeOn prescribes a set of guidelines for multiple alternative processes for various stages of ontology development that may change with the design decisions.

4.2 Methodology

The end goal of DOOP is the creation of a tool, e.g. a browser extension, that uses the ontology as a knowledge base to parse online privacy policies and highlight sections in the policy that would address a user’s concerns. The development of this is beyond the scope of this thesis; hence, only the ontology is presented here. A hybrid approach was used for constructing DOOP. NeOn was used for the structuring of the requirements and parts of the ontology that Ontology 101 could not do on its own, and Ontology 101 was used everywhere but the initial stages of ontology creation. The rest of this section is organized as such:

1. First an overview of DOOP is provided in Section 4.2.1
2. Then the purpose of building DOOP is clarified in Section 4.2.2

4.2.1 Overview

DOOP forms the backbone of a tool that semi-automates the evaluation of online privacy policies and helps answer key user concerns related to online privacy. The ultimate goal was the reduction of the amount of text users had to read to know whether their concerns were being met or not. Hence, utilizing the CQs, DOOP needed to capture most terms in the online privacy domain and relate to them back to the CQs and the privacy categories, i.e., data collection, data retention, data security, data sharing, target audience, user access, and user choice. This had to be done in a way such that recommendations could be made by the tool to the user regarding CQs whose related categories are overlapping the original enquiry (CQ). The goal of building DOOP was to support the following end users in particular:

1. Privacy researchers.
2. NLP experts who are interested in doing work in the online privacy domain.
3. Software developers who would like to use an ontology to create tools for the online privacy domain.

4.2.2 Purpose

The primary goal of DOOP was to capture the vocabulary found in online privacy policies, and map relationships between the terms and concepts. In order to associate the terms with general context in which they are used, they must be mapped to privacy categories. In addition, the ontology must also be able to capture and suggest concerns that the user might be interested in based on their primary concern. To facilitate this, the ontology must also store and map CQs, in plain English and its associated SPARQL query (SPARQL Protocol and RDF Query Language), to the terms. Lastly, the ontology must be able to capture information about the basic structure of the policies, i.e., meta-data about the privacy policies including cookie policies. Cookie policies, though usually part of privacy policies, are sometimes stored separately. E.U. guidelines on privacy and data protection requires all websites that do business within the E.U. to have a cookie policy ([Information Providers Guide: Cookies](#), n.d.). Hence, it was necessary to accommodate cookie policies separately as they can be updated separately.

4.2.3 Intended Use Cases

The ontology is intended for use for the following use cases:

1. **Vocabulary** - The ontology should be able to contain important terms, their synonyms and acronyms, their equivalents, and have them classified within categories.

2. **Retrieval** - The ontology should make it possible to retrieve terms, along with their synonyms, acronyms, and equivalents, through a query for specified classification or question, i.e., privacy category or CQ. It must also allow the retrieval of meta-data pertaining to the privacy policies stored in the ontology.

3. **Organization** - The ontology should provide a logical manner of classifying new terms, and adding new categories for classification. It should further ensure that meta-data for unseen privacy policies can be easily added; and also new questions (CQs).

4. **Recommendation** - Based on the mappings between terms to categories and terms to questions, the ontology should be able to recommend other categories and questions that might be of interest due to overlapping terms.
4.2.4 Ontology requirements

We propose that DOOP should satisfy the following requirements:

1. The ontology must satisfy all 3 CQs described in Section 4.2.7.

2. The ontology must also store the CQs in both English and SPARQL formats, and be associated with the various privacy categories same as the keywords.

3. The ontology must also be able to save meta-data related to privacy policies.

4.2.5 Construction

For the construction of DOOP, a hybrid approach was used: a combination of Ontology 101 and NeOn methodologies in conjunction with iterative development. Ontology 101 was proposed with the intent of using Protégé (see Section 4.1.3) for the construction of ontology. Since the latest version of Protégé (Stanford Center for Biomedical Informatics Research, 2017) was used in the construction of DOOP, Ontology 101 was used as the prime methodology. Furthermore, ontology engineering guidelines provided under NeOn’s Scenario 1 (From Specification to Implementation), which includes steps from other methodologies, were used as the foundation to specify and build DOOP. Instead of building a complete ontology that exhaustively considers every possible case that should be covered, DOOP was built in iterations as described by RapidOWL. By expanding the vocabulary one query at a time the ontology remains open and malleable enough such that future developments require relatively less effort to alter the structure of the ontology as needed. This ensures that DOOP remains a multipurpose domain ontology rather than a task-specific one.

The following goals were kept in mind to ensure ease of development and project longevity:

1. The ontology must be implemented using freely available tools so that it can be easily extended.

2. The ontology must be implemented in a widely used ontology machine language which has freely available libraries so that it can portable to be implemented as part of a larger tool, and be extended automatically within the tool.

For these goals, the ontology was implemented in OWL-DL using the Protégé tool (version 5.2.0) for ontology engineering.

4.2.6 Source for DOOP Development

There are inconsistencies in the language used between privacy policies [1.1]. These inconsistencies meant that the corpus of privacy policies used for ontology development had to be large and diverse to capture as many terms as possible, and from different economic...
zones. For these reasons, the corpus of privacy policies used in Chapters 2 and 3 was used as the source for the key terms.

The key terms found in those Chapters largely became individuals (instances) in the ontology, with a few being used as classes. Seven classes were identified: data collection, data retention, data security, data sharing, target audience, user access, and user choice. These were identified in the work done by (Guntamukkala et al., 2015); and were based on the logical division of policies described under the FIPPs, and principles identified under OECD’s guidelines for protection of privacy and data flows (OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data, n.d.). These classes are commonly found in both cookie and privacy policies. The hierarchy of the ontology was developed as needed to satisfy the CQs.

### 4.2.7 Competency questions

Competency questions are a set of queries that the ontology should be able to answer based on its axioms (Section 1.2.3). This is why they are used for not only defining ontology requirements but also ontology evaluation; the result from a CQ can be used to determine the correctness of an ontology. CQs can be used for evaluation either manually or automatically through the use of SPARQL queries. DOOP was constructed and evaluated through the use of CQs. After defining a CQ, the ontology was iteratively constructed by defining as many axioms needed to answer the CQ. Of the most universal questions that are answered by online privacy policies, 3 CQs were selected for constructing the ontology for proof-of-concept (Table 4.1). The 3 CQs are described below, along with their: natural language query, answer, and SPARQL query.

1. **Does this website share my personal information with third-parties?**
   - **Query:** What are all of the terms related to personal information and third-parties?
   - **Answer:** email address, pii, name, ad networks, advertiser, advertizer, personal information, first name, middle name, last name, age, personally identifiable information, mother’s maiden name.
   - **Category:** Data Sharing
   - **SPARQL Query:**

     ```sparql
     PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
     PREFIX owl: <http://www.w3.org/2002/07/owl#>
     PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
     PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
     PREFIX op: <http://www.semanticweb.org/dhiren/ontologies/2016/7/online_privacy#>
     
     SELECT ?term
     WHERE {
       ?term op:hasProperty ?property.
       FILTER (regex(?term2, "personal information") OR regex(?term2, "third-parties"))
     }
     ```
Table 4.1: 17 concerns that users (children, teenagers, adults, seniors) most expressed about their online privacy.

<table>
<thead>
<tr>
<th>Privacy Category</th>
<th>User Concern</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>User Access / Choice &amp; Purpose</strong></td>
<td>1. What countries does this privacy policy apply to?</td>
</tr>
<tr>
<td></td>
<td>• Does this policy apply to me where I am right now?</td>
</tr>
<tr>
<td></td>
<td>2. Does this policy have a cookie policy?</td>
</tr>
<tr>
<td></td>
<td>• Does this website use tracking cookies?</td>
</tr>
<tr>
<td></td>
<td>• What are they doing with cookies?</td>
</tr>
<tr>
<td></td>
<td>3. With what laws/acts does this policy comply with?</td>
</tr>
<tr>
<td></td>
<td>4. Does this policy include provisions for children's safety? For example, Children’s Online Privacy Protection Act (COPPA).</td>
</tr>
<tr>
<td></td>
<td>5. Will the user be notified of any changes to this policy?</td>
</tr>
<tr>
<td><strong>Data Collection</strong></td>
<td>1. Which PII are covered by this policy?</td>
</tr>
<tr>
<td></td>
<td>2. Does this policy have an opt-out policy?</td>
</tr>
<tr>
<td></td>
<td>3. Does this policy have an opt-in policy?</td>
</tr>
<tr>
<td></td>
<td>4. Does this website track me?</td>
</tr>
<tr>
<td></td>
<td>5. Does this website store cookies on my computer?</td>
</tr>
<tr>
<td></td>
<td>6. Can I request the website to not track me?</td>
</tr>
<tr>
<td><strong>Data Security</strong></td>
<td>1. Is the connection between my browser and the website encrypted?</td>
</tr>
<tr>
<td></td>
<td>2. Is my PI encrypted?</td>
</tr>
<tr>
<td><strong>Data Sharing</strong></td>
<td>1. Does this policy describe how PII is being shared?</td>
</tr>
<tr>
<td></td>
<td>2. Which PII are being shared under this policy?</td>
</tr>
<tr>
<td><strong>Data Retention</strong></td>
<td>1. Does this policy specify any data retention and storage practices?</td>
</tr>
<tr>
<td></td>
<td>2. Can I request that my PI be deleted or removed?</td>
</tr>
</tbody>
</table>
2. Does this website use tracking cookies?

- **Query:** What are all of the terms related to tracking and cookies?
- **Answer:** do not track, tracking cookie, web beacon.
- **Categories:** data collection, user choice
- **SPARQL Query:**

```sparql
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX op: <http://www.semanticweb.org/dhiren/ontologies/2016/7/online_privacy#>

SELECT ?term
WHERE {
    { ?term a op:Tracking; op:relatedTo op:DataCollection }
UNION
    { ?term a op:Tracking; op:relatedTo op:UserChoice }
UNION
    { ?term a op:Cookie; op:relatedTo op:DataCollection }
UNION
    { ?term a op:Cookie; op:relatedTo op:UserChoice }
}
```

Listing 4.2: SPARQL query for returning terms from DOOP that are related to tracking and cookies.

3. Can I opt-in/opt-out of information gathering?

- **Query:** What are all of the terms related to opt-in and opt-out policies?
- **Answer:** opt-in, opt-out, consent.
- **Categories:** data collection, user choice
- **SPARQL Query:**

```sparql
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX op: <http://www.semanticweb.org/dhiren/ontologies/2016/7/online_privacy#>

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX op: <http://www.semanticweb.org/dhiren/ontologies/2016/7/online_privacy#>

SELECT ?term
WHERE {
    { ?term a op:Tracking; op:relatedTo op:DataCollection }
UNION
    { ?term a op:Tracking; op:relatedTo op:UserChoice }
UNION
    { ?term a op:Cookie; op:relatedTo op:DataCollection }
UNION
    { ?term a op:Cookie; op:relatedTo op:UserChoice }
}
```

Listing 4.2: SPARQL query for returning terms from DOOP that are related to tracking and cookies.
SELECT ?term
WHERE {
  { ?term a op:OptIn; op:relatedTo op:DataCollection } UNION
  { ?term a op:OptIn; op:relatedTo op:UserChoice } UNION
  { ?term a op:OptOut; op:relatedTo op:DataCollection } UNION
  { ?term a op:OptOut; op:relatedTo op:UserChoice }
}

Listing 4.3: SPARQL query for DOOP to return terms related to opt-in and opt-out.

4.2.8 Ontology specific classes

DOOP specific classes and related definitions were entered manually into Protégé in a top-down development process. The most general classes were defined first moving progressively to specialized classes which became sub-classes. Since all of the classes were unique, Unique Resource Identifiers (URIs) were created for all of the classes. The preferred names of all classes are in singular noun form. Descriptions of the classes, and their alternative names (if any), were asserted as annotations using the ‘dc:description’ and ‘dc-terms:alternative’ annotation properties respectively. To allow for future development of internationalization, English was selected as the language for all annotations.

4.3 Results

In this section, the characteristics of DOOP are described in Sections 4.3.1, 4.3.2, 4.3.3, 4.3.4 and the results from ontology validation is presented in Section 4.3.5. A breakdown of DOOP is given in Table 4.2. An example of a class is given in Table 4.3.

Table 4.2: Breakdown of DOOP.

<table>
<thead>
<tr>
<th>Property</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Axioms</td>
<td>304</td>
</tr>
<tr>
<td>Logical axiom count</td>
<td>171</td>
</tr>
<tr>
<td>Declaration axioms count</td>
<td>71</td>
</tr>
<tr>
<td>Class count</td>
<td>15</td>
</tr>
<tr>
<td>Object property count</td>
<td>6</td>
</tr>
<tr>
<td>Data property count</td>
<td>10</td>
</tr>
<tr>
<td>Individual count</td>
<td>35</td>
</tr>
<tr>
<td>Annotation property count</td>
<td>8</td>
</tr>
</tbody>
</table>
4.3.1 Top structure

The ontology is divided into four main classes derived from the standard OWL root class for everything *owl:Thing*: Keyword, PrivacyCategory, PrivacyPolicy, and Question. The following are rules for creating the rest of the classes and individuals. Refer to Figures 4.2 and 4.3.

1. The Keyword class captures most of the vocabulary contained in DOOP. As shown in Figure 4.2 it is a sub-class of *owl:Thing*. Classes act as sets of individuals, hence, Opt-in and Opt-out form instances of Keyword (Fig. 4.3) but Cookie is a sub-class of Keyword; this is because the class Cookie has the instances, ‘Do Not Track’ and ‘Web Beacon’, which are types of cookies. Legal Act, Country, and Organization, are all sub-classes of Keyword which are interrelated by relationships: ‘Applies To’, ‘Operates In’, ‘Enacted’, and ‘Based’. Legal acts ‘Applies To’ country which is an inverse of ‘Enacted’. Similarly, organizations ‘Operates In’ countries, and countries serve as ‘Base’ for organizations which is also an inverse relationship to the former relationship.

2. Privacy category class does not have a sub-class, but has 7 individuals: data collection, data retention, data security, data sharing, target audience, user access, and user access. Both Keyword and Question classes share a ‘Related To’ relationship with Privacy Category, since keywords and questions are logically classified under various privacy categories. An example of this is shown in Figure 4.3 where Opt-Out is ‘Related To’ User Choice, Data Collection, and Data Sharing. Since, Opt-Out is ‘Similar To’ Opt-In, it is inferred that it too is related to the three categories.

3. The Question class is where all of the questions are stored as individuals. This class has no sub-class because all of the questions are types of Question. Additional information is stored as object properties, e.g., concern, competency question, and SPARQL query.

4. The Privacy Policy class stores meta-data for privacy policies already processed. It is divided into two sub-classes: Policy Document and Cookie Policy. Privacy policies are individuals of the Policy Document class, and if a separate cookie policy exists then its meta-data is stored under the Cookie Policy class.

<table>
<thead>
<tr>
<th>Preferred Name</th>
<th>Personally Identifiable Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alternative</td>
<td>PII</td>
</tr>
<tr>
<td>Description</td>
<td>Information about an individual person, organization, or any other entity.</td>
</tr>
<tr>
<td>SubClass Of</td>
<td>Keyword</td>
</tr>
<tr>
<td>Instances</td>
<td>Email Address, Email Preference, Name, Age, PII</td>
</tr>
</tbody>
</table>
Figure 4.2: Top class structure of DOOP. Some classes and individuals are not shown due to space limitations.

Figure 4.3: Example of individuals in DOOP. Some classes and individuals are not shown due to space limitations.
Table 4.4: Object properties presently available in DOOP.

<table>
<thead>
<tr>
<th>Relationship</th>
<th>Domain</th>
<th>Range</th>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>is a superclass of</td>
<td>∞</td>
<td>∞</td>
<td>Transitive</td>
</tr>
<tr>
<td>applies to</td>
<td>LegalAct</td>
<td>Country</td>
<td>Functional, InverseOf:enacted</td>
</tr>
<tr>
<td>based</td>
<td>Country</td>
<td>Organization</td>
<td>InverseOf:OperatesIn</td>
</tr>
<tr>
<td>enacted</td>
<td>Organization</td>
<td>LegalAct</td>
<td>InverseOf:AppliesTo</td>
</tr>
<tr>
<td>operates in</td>
<td>Country</td>
<td>Country</td>
<td>Functional, InverseOf:Based</td>
</tr>
<tr>
<td>related to</td>
<td>Keyword, Question</td>
<td>PrivacyCategory</td>
<td>Symmetric</td>
</tr>
<tr>
<td>similar to</td>
<td>Keyword</td>
<td>Keyword</td>
<td>Transitive, Symmetric</td>
</tr>
</tbody>
</table>

4.3.2 DOOP defining classes

The general structure of the ontology was developed with the firm intention of developing a tool to dissect an online privacy policy into sections that the user might be most interested in based on their concerns and the captured vocabulary offered by the domain ontology. Since, to the best of our knowledge, no previous attempt had been made to capture only the vocabulary in online privacy policies, there were no ontologies to refer to for creating classes and structure. PrivOnto created by CMU is also classified as a domain ontology, however, due to the purpose and methodology employed by the ontology engineers its class structure was unhelpful in the creation of DOOP. The limitation of PrivOnto is discussed in Section 1.1.

4.3.3 Inference and structure

DOOP is primarily composed of single is-a asserted inheritance structure, expressed with subclass relations in OWL-DL. However, other relations also exist to enable further development and capturing of more complex logic. An exhaustive list of object relations along with their properties is described in Table 4.4. These provide useful classification hierarchies and extendability for the users of the ontology. These relationships allow the user to infer: which keywords are related to what privacy category and question; which questions share a certain number of keywords, useful of recommendation; what legal acts are enforceable in a country; what organizations enforce which legal acts and where they are located; determine the overlap of vocabulary between privacy categories. DOOP is consistent with all three reasoners in Protégé 5.2.0: FaCT++ (Tsarkov & Horrocks, 2006), HermiT (Glimm, Horrocks, Motik, Stoilos, & Wang, 2014), and Pellet (Sirin, Parsia, Grau, Kalyanpur, & Katz, 2007).
4.3.4 General class axioms

DOOP contains some general class axioms for the categorization and saving of meta-data about privacy/cookie policies in the ontology. Data properties such as date, retrieval date, the body text, and URL, along with the object properties helps to extend the usefulness of the ontology beyond just capturing the vocabulary. It allows the creation of relationships between questions and preprocessed parts of the privacy policy. This would drastically reduce the time needed to process each policy as it acts as a form of caching. DOOP also allows the storing of questions within the ontology. This is especially useful, as vetted SPARQL queries can be saved directly into the ontology so that developers of tools using the ontology would get a set of canned queries for quick development. Furthermore, since the questions are related to the keywords, users of the ontology can infer other queries that they might be interested in by looking at the overlap of the set of keywords related to questions.

4.3.5 Validation

Owing to the individual and subjective nature of ontologies, ontology validation is a difficult process (Sec. 1.2.3). DOOP was validated in two ways: CQs (Sec. 4.2.7) and data driven (Sec. 1.2.3). CMU’s OPP-115 (Sec. 1.4.3) data set was used as the primary benchmark for validation. Since there were 10 policy annotators, there were many overlaps in the labelling of policies, as well as multiple labels for the same policy. To reduce redundancy, the authors of OPP-115 consolidated annotations with three convergence thresholds: 0.5, 0.75, and 1. These were calculated as normalized aggregated overlap of spans of text assigned the same data attribute for the same data practise identified by multiple annotators. For our validation, annotations from all three thresholds were considered. Furthermore, OPP-115 annotators have classified their annotation under 10 categories which had to be mapped to our 7 categories; Figure 4.4 shows this mapping. The mapping was necessary because at the time of undertaking this research OPP-115 data set had not yet been released so the categories introduced by (Guntamukkala et al., 2015) were used. For validation, four experiments were conducted that evaluated various aspects of the ontology:

1. **Correctness**: Compute the number of matched privacy categories for the same sentences from both DOOP (based on the keywords the sentence contains) and OPP-115.

2. **Policy coverage**: Compute the the number of sentences that the reader has to theoretically read to understand the risks associated with his concerns.

3. **Completeness**: Compute per policy keywords that existed in the ontology but not in the policy.

4. **Correctness**: Compute cases where the keyword’s assigned category in the ontology did not match the OPP-115’s annotation’s assigned category.
Experiment 1

In order to compare categories, the annotation set’s results had to be processed. The results are presented as a CSV file for each privacy policy in the consolidation directory as indicated by their manual. The column description used by the CSV files is presented below:

A) annotation ID (a globally unique identifier for a data practice)
B) batch ID (name of a batch in the annotation tool; often indicates who the annotators were)
C) annotator ID
D) policy ID (this corresponds to the numeric prefixes in the policy filename, as found in other directories)
E) segment ID (the zero-indexed, sequential identifier of the policy segment; e.g., the first segment in a policy’s text is segment zero)
F) category name
G) attribute-value pairs (represented as JSON, this where the annotations are stored)
H) policy URL
Table 4.5: Results for DOOP validation, experiment 1.

<table>
<thead>
<tr>
<th>Query</th>
<th>0.5</th>
<th>0.75</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SM</td>
<td>PM</td>
<td>%</td>
</tr>
<tr>
<td>1</td>
<td>550</td>
<td>3418</td>
<td>62.13%</td>
</tr>
<tr>
<td>2</td>
<td>341</td>
<td>308</td>
<td>90.32%</td>
</tr>
<tr>
<td>3</td>
<td>848</td>
<td>648</td>
<td>76.42%</td>
</tr>
</tbody>
</table>

Table 4.6: Results for DOOP validation, experiment 2.

<table>
<thead>
<tr>
<th>Query</th>
<th>0.5</th>
<th>0.75</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TS</td>
<td>SS</td>
<td>Coverage</td>
</tr>
<tr>
<td>1</td>
<td>91.7</td>
<td>22.99</td>
<td>30.99%</td>
</tr>
<tr>
<td>2</td>
<td>91.7</td>
<td>1.18</td>
<td>1.42%</td>
</tr>
<tr>
<td>3</td>
<td>91.7</td>
<td>4.12</td>
<td>4.32%</td>
</tr>
</tbody>
</table>

I) date

A qualified positive match of categories occurs when the selectedText for an annotation under attribute-value contains any of the keywords returned by DOOP for a query whose DOOP categories also match their mapped OPP-115 category name. Results for all three queries, and for all three thresholds are presented in Table 4.5. A sentence in a policy that contains any of the keywords that DOOP returns is denoted by SM or Sentences Matched, and sentences for which there is category match is denoted by PM or Positive Matched, while score is the percentage.

Experiment 2

For this experiment, the average number of sentences that exist in a policy was first calculated, then it was divided by the number of sentences that contained the keywords that were returned after the execution of the SPARQL query for a privacy concern. The results for all queries as well as for all convergence thresholds are presented in Table 4.6. Average number of total sentences in privacy policies is denoted by TS, whereas selected sentences that contain a keyword also in the returned query from DOOP is denoted by SS.

Experiment 3

In this experiment, the number of keywords that the ontology returned for a particular query that did not exist in the privacy policies was calculated. The primary purpose here was to investigate how many unique terms existed in DOOP that did not exist in the policy. Since all of 115 policies used in OPP-115 were American, unique terms found in DOOP
would indicate a geographic non-specific ontology that can be generally used in most English speaking countries. Results from this experiment are reported in Table 4.7. Average number of keywords found is denoted by KF, and average number of keywords not found is denoted by NF.

### Experiment 4

Since, assigning categories to keywords is a manual task, and privacy categories from DOOP were further mapped onto category names assigned by annotators from OPP-115, we wanted to know how often we differed in opinion. Thus, in this experiment we investigated how often keyword assigned category in DOOP differs from OPP-115. Results from this experiment is presented in Table 4.8.

### 4.4 Discussion

Of the 115 policies used for the evaluation, the longest policy had 293 sentences, and the least had 4. There were two policies that had 4 sentences: [tangeroutlet.com](http://tangeroutlet.com) and [solarviews.com](http://solarviews.com). On inspection, Solar Views site did have only 4 sentences in its policy:

```
We allow third-party companies to serve ads and/or collect certain anonymous information when you visit our web site. These companies may use non-personally identifiable information (e.g., click stream information, browser type, time and date, subject of advertisements clicked or scrolled over) during your visits to this and other Web sites in order to provide advertisements
```

Table 4.7: Results for DOOP validation, experiment 3.

<table>
<thead>
<tr>
<th>Query</th>
<th>0.5</th>
<th>0.75</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>KF</td>
<td>NF</td>
<td>Unique</td>
</tr>
<tr>
<td>1</td>
<td>3.86</td>
<td>9.14</td>
<td>70.31%</td>
</tr>
<tr>
<td>2</td>
<td>0.41</td>
<td>2.59</td>
<td>86.33%</td>
</tr>
<tr>
<td>3</td>
<td>1.39</td>
<td>1.61</td>
<td>53.67%</td>
</tr>
</tbody>
</table>

Table 4.8: Results for DOOP validation, experiment 4.

<table>
<thead>
<tr>
<th>Query</th>
<th>0.5</th>
<th>0.75</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total</td>
<td>Diff</td>
<td>%</td>
</tr>
<tr>
<td>1</td>
<td>5501</td>
<td>2083</td>
<td><strong>37.87%</strong></td>
</tr>
<tr>
<td>2</td>
<td>385</td>
<td>84</td>
<td><strong>21.82%</strong></td>
</tr>
<tr>
<td>3</td>
<td>1696</td>
<td>1048</td>
<td><strong>61.79%</strong></td>
</tr>
</tbody>
</table>
about goods and services likely to be of greater interest to you. These compa-

nies typically use a cookie or third party web beacon to collect this informa-

tion. To learn more about this behavioral advertising practice or to opt-out of
this type of advertising, you can visit networkadvertising.org.

Tanger Outlet had far more, but it was not picked up by the page scraping script as it was
hidden under a dropdown on the page, (see Fig 4.5). The OPP-115 data set was used in
good faith, and extensive checking and cleaning was not performed. On average there were
25 sentences per policy (Fig. 4.7) and 28 words per sentence (Fig. 4.6).

In Experiment 1 (Table 4.5), a mean of 76.07% match for privacy categories with a stan-
dard deviation of 12.62 was achieved. Since the OPP-115 data set was manually curated
by domain experts, a high degree of match means a high degree of accuracy achieved by
the ontology for identifying sentences in context based on the vocabulary. Now, the users
need not read the entire policy, but can be directed to appropriate sentences in the policy
that deal directly with their concerns with a reasonable amount of accuracy. Additionally,
there is a negligible increase in the accuracy of the automatic categorization when the con-
vergence threshold is 0.75. This could be as consolidation reduces redundancy, without
overdoing it at 1.0 convergence threshold.

One of the prime reasons that users do not read privacy policies, and are left uninformed,
is that they tend to be overly long. Any tool that is trying to fix this issue must not only
find correct information but also require less reading on the part of the user. In Experi-
ment 1, algorithmic assignment of privacy categories to sentences was performed favourably
against the manual annotations performed by domain experts. Thus, in Experiment 2,
policy coverage was investigated to identify how much of a policy is the user asked to read
for the three identified concerns. This experiment demonstrated (Table 4.6) that the user
has to read on average 10.54 sentences with a standard deviation of 11.99, or about 12.24%
Figure 4.6: Privacy policies: sentences versus words.

Figure 4.7: Privacy policies: average lengths of privacy policies.
Query 1: OPP-115 Threshold 0.75

Figure 4.8: Sentences selected for reading for query 1: ‘Does this website share my personal information with third-parties?’.
Query 2: OPP−115 Threshold 0.75

Figure 4.9: Sentences selected for reading for query 2: ‘Does this website use tracking cookies?’.
Figure 4.10: Sentences selected for reading for query 3: ‘Can I opt-in/opt-out of information gathering?’.
of a policy with a standard deviation of 14.12 to know if all of their concerns are being met. Assuming that a paragraph is roughly 10 sentences, then based on research done by (Ramanath et al., 2014) (Sec.1.1), we know that it would take roughly 45 seconds to read it. This reduced time makes the privacy policies more user friendly and should encourage more users to read policies even if partially.

To further investigate these results, additional analysis of the individual results from experiments for all of the thresholds and queries was conducted. Figures 4.8, 4.9, and 4.10 show the results of the queries for the 0.75 threshold, the other two are included under Appendix A. A radical relationship between the amount of sentences selected and the length of the policies was expected, where the reading proportionally increases as the length of the policies increase, but then stabilize at some horizontal asymptote. However, this did not occur. A strong positive linear relationship was observed for the first experiment (Fig. 4.8), no correlation for the second experiment (Fig. 4.9), and weak positive for the third experiment (Fig. 4.10). A qualitative analysis provided several clues for these behaviour:

1. The vocabulary in the first query, was trying to capture more than one concern. Since the ontology only returned a vocabulary of terms, it was hard to determine the correct context sometimes as one set of keywords could be used in multiple instances under different contexts. One possible solution to this problem could be having the ontology also capture POS tags that determines the structure of the sentences and identifies the associated verb (e.g. sharing) that would enforce a context under which the sentence occurs. This would help distinguish one context from another where the most of the vocabulary is shared.

2. Policies were repeated throughout the document. This accounted for the linear relationship for the first experiment. Redundancies in the policies drove up the number of sentences to read.

3. Keywords returned by the ontology were narrowly defined. This was an important distinction with the second query regarding tracking cookies. The keyword ‘cookie’ was not being used because not all cookies are tracking, this meant that several cases where that term was being used to establish context was not captured. For example, “We do not use tracking cookies.” would be selected, but, ‘Cookies may be used to track you’, would be ignored. Similar to the first problem, POS tags for some terms could be captured by the ontology to identify context as a remedy to this problem. In the OPP-115 data set, only 30% of the policies had a ‘tracking cookie’ policy that was part of the privacy policy that was extracted. For comparison, the term ‘cookie’ was added to the list of keywords for the second query, and all of the queries re-run; results are presented in Section 4.4.1.

4. Some policies were missing entirely. Sometimes, a supplementary document was used to state policies, e.g., ‘Cookie Policy’. This supplementary document was not stored on the same page as the privacy policy; hence, it was picked up the scraper scripts. This a difficult challenge to solve, as there is no consensus as to what the URL must be for the cookie policy. However, a reasonable attempt can be made to scrape this page as well and amend it to the privacy policy.
In the creation of the taxonomy for DOOP, the vocabulary was not restricted to a particular geographically intended audience, in order to make the ontology as general purpose as possible. OPP-115’s data set contained only American privacy policies. Hence, there were terms in DOOP that did not exist in OPP-115. Experiment 3 was conducted to investigate the uniqueness of DOOP in comparison to OPP-115. In Table 4.7, observe that on average 70.10% of terms are unique to DOOP with a standard deviation of 14.14. This was expected, as not just the American policies were considered when extracting keywords from privacy policies as explained in Chapters 2 and 3, but also Canadian and European ones. The 70% terms also include localization of the American spelling along with synonyms, hyponyms, and E.U. and Canada specific terms, which made the ontology more unique here, e.g., advertiser/advertizer, and name/full name.

Finally, in Experiment 4 investigated how much the labels of keywords agreed between DOOP and OPP-115. The mean disagreement between the data sets was 40.97% with a standard deviation of 17.20. The most disagreement being with query 3. One of the reasons for this discrepancy could be due to the mapping of categories from OPP-115 to DOOP. Since the mapping of the privacy categories between the data sets was not one-to-one; approximations had to be made. This meant that one category in one data set was mapped on to multiple categories in the other introducing a great amount of variance in the topics captured by each category. Another explanation has to do with the limited amount of vocabulary DOOP currently captures. In its present state, it was created to be a proof-of-concept system. As the vocabulary increases, it is expected that the results for all four experiments would also improve.

### 4.4.1 Generalizing keyword searches

Investigation was conducted to see what happens when more generic keywords are added to queries for focused and narrowly defined queries, such as query 2. The term ‘cookie’ was added to the list of keywords for the second query, and all of the experiments re-run. The results are shown in Tables 4.9, 4.10, 4.11, 4.12 and Figure 4.11.

In general, the total number of sentences dramatically increases from 341 to 1442 (Table 4.9), and the accuracy suffers as well from 90% to 85%. This is expected as ‘cookies’ is mentioned more often because they are used for more than just tracking. They are also widely used for storage of temporary data. This can be also observed in Table 4.11 which shows that there is at least one word common to the vocabulary in the ontology and is consistently being found in the policies. Furthermore, this results in the increase of
Table 4.10: Results for DOOP validation query 2, experiment 2.

<table>
<thead>
<tr>
<th>‘Cookie’</th>
<th>0.5</th>
<th></th>
<th>0.75</th>
<th></th>
<th>1</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TS</td>
<td>SS</td>
<td>Coverage</td>
<td>TS</td>
<td>SS</td>
<td>Coverage</td>
</tr>
<tr>
<td>No</td>
<td>91.7</td>
<td>1.18</td>
<td>1.42%</td>
<td>91.7</td>
<td>1.18</td>
<td>1.42%</td>
</tr>
<tr>
<td>Yes</td>
<td>91.7</td>
<td>8.12</td>
<td>9.69%</td>
<td>91.7</td>
<td>8.12</td>
<td>9.69%</td>
</tr>
</tbody>
</table>

Figure 4.11: Results for experiment 2 for query 2 after adding ‘cookie’ to the keywords.
Table 4.11: Results for DOOP validation for query 2, experiment 3.

<table>
<thead>
<tr>
<th>‘Cookie’</th>
<th>0.5</th>
<th>0.75</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>KF</td>
<td>NF</td>
<td>Unique</td>
</tr>
<tr>
<td>No</td>
<td>0.41</td>
<td>2.59</td>
<td><strong>86.33%</strong></td>
</tr>
<tr>
<td>Yes</td>
<td>1.04</td>
<td>2.96</td>
<td><strong>74.00%</strong></td>
</tr>
</tbody>
</table>

Table 4.12: Results for DOOP validation for query 2, experiment 4.

<table>
<thead>
<tr>
<th>‘Cookie’</th>
<th>0.5</th>
<th>0.75</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total</td>
<td>Diff</td>
<td>%</td>
</tr>
<tr>
<td>No</td>
<td>385</td>
<td>84</td>
<td><strong>21.82%</strong></td>
</tr>
<tr>
<td>Yes</td>
<td>1486</td>
<td>482</td>
<td><strong>32.44%</strong></td>
</tr>
</tbody>
</table>

estimated number of sentences to read per policy on average from 1.18 to 8.12 (Table 4.10). The addition of generic keyword also increased the variance of the sentences to read, as can be observed in Figure 4.11. This has also led to fewer policies being flagged as having 0 sentences to read. Once again, the number of sentences in a policy and the amount of reading a user has to do is linearly correlated. One of the most important measure is the increase of disagreement between the recommended sentences versus the annotated ones (Table 4.12). This indicates that adding generic terms deteriorates the overall quality of the recommendations. The indicators in this short study demonstrate that in order for the recommended reading to be useful to the user, it must have fewer generic terms and more targeted ones.
Chapter 5

Conclusion

Privacy policies play an important part in informing users about their privacy concerns. As the world becomes more interconnected and online, security threats prompt users to become more privacy aware, making online privacy policies the primary documents for users making informed decisions. These policies are long and difficult for most users to understand and are infrequently read, presenting a challenge for users. Previous attempts at creating machine readable policies have had limited success as it places the onerous task of crafting these polices on businesses.

This thesis proposed a novel approach to reducing the amount of text a user has to read by using a domain ontology and NLP to identify key areas of the policies that the user should read to address their concerns and take appropriate action. The approach consisted of constructing DOOP, a domain ontology for online privacy policies, validated against CMU’s OPP-115 data set of annotated policies by domain experts.

DOOP was produced in three phases: keywords gathering, taxonomy building, and adding relationships between the queries and the taxonomy. In the first phase, supervised and unsupervised learning keyword extraction algorithms were evaluated. The algorithms were applied to a corpus of 631 online privacy policies from websites of various industries. To evaluate the efficacy of these algorithms, a manually extracted data set of keywords was created from 21 policies. Of the four unsupervised algorithms, TF-IDF, RAKE, TextRank, and AlchemyAPI, TF-IDF performed best over a smaller corpus with an $F_1$-score of 27%, and AlchemyAPI performed best over the entire corpus with an $F_1$-score of 5%.

By comparison, supervised learning algorithm, KEA, performed slightly better than the best unsupervised algorithm (AlchemyAPI) for the entire corpus with an $F_1$-score of 7%. Keywords extracted by KEA were added to the list of keywords manually extracted. Using the three queries identified in Chapter 4, keywords from this list were shortlisted to determine which keywords best represented those queries. These keywords were then mapped to the various privacy categories, and also the queries which were also stored in DOOP.

The primary goal of this research was to reduce the burden of having to read long, jargon-filled privacy policies by directing users to relevant parts of the policies for a given
question. The approach described in this thesis when applied to the privacy policies in the OPP-115 data set resulted in 69%, 99%, and 96% reductions in reading for the 3 sample questions, and on average it would take about 45 seconds to read the relevant sentences (11 on average). By comparison, the average time to read privacy policies is estimated to be 8 – 12 minutes [McDonald & Cranor 2008]. Reducing the reading time should encourage more users to read privacy policies and make informed decisions online.

In addition, DOOP is the first ontology to capture the vocabulary of online privacy policies. It also provides a method to describe the vocabulary in terms of the privacy categories that are widely used by Federal Trade Commission, Canada Privacy Commissioner, and identified by the directives proposed by the European Commission.

Furthermore, the vocabulary is mapped to the queries, which are also stored in the ontology. This allows ontology developers to propose additional insight into related queries and their associated vocabulary. Moreover, in Experiments 2 and 3 showed the difficulty of term extraction from privacy policies using unsupervised learning. Experiment 3.3 improved on that by employing supervised learning algorithms.

Through the user experiments, the ease of expanding the vocabulary of online privacy by employing and training non-domain experts was demonstrated. When used in concert with supervised learning algorithms, this proved to be a cost-effective way of expanding the vocabulary in the ontology. The development of DOOP showed the usefulness of domain ontologies when applied to privacy policies, and also demonstrated a cost-effective way of maintaining and expanding it in the future.

5.1 Limitations & Future Work

In the domain of online privacy policies, ontologies with NLP achieved valuable results. However, there are some limitations:

1. In its present state, DOOP only covers 3 of the 15 CQs identified. This was a design decision needed to fulfill time constraints, and a way of showing the worthiness of DOOP. In the future, this should be expanded to all 15 CQs to capture a wide array of user concerns.

2. DOOP has limited vocabulary. As previously stated, DOOP only covers 3 CQs. As the number of CQs increases the vocabulary will grow to accommodate more terms capturing all of the concerns. Term extraction experiments saw several hundred terms marked as important. Only a fraction of these were included in the ontology. All of these will need to be incorporated into the ontology to create a richer taxonomy.

3. Not all of DOOP’s features were validated as several features of the ontology were supplementary and hence, not validated. These features include: recommending other concerns that the user might have; manually validating the structure of the
ontology by the use of domain experts; and accessing and adding meta-data for privacy policies. Future evaluations of the ontology should include the testing of these features.

4. DOOP currently performs poorly in capturing context as shown by the ‘tracking cookies’ problem. One possible solution could involve adding an object property to the individuals with a POS tag stating how the term is being used in a sentence. When parsing a document for these terms, sentences can be tagged with a POS tagger and the tag cross-referenced with the object property. This would help capture the context through word usage.

5. To reduce costs, keyword extraction algorithms were used to augment manual term extraction. Manual extraction is expensive and time consuming when dealing with a large number of privacy policies. Experiments 2 and 3 showed that keyword extraction algorithms are not very effective with privacy policies. Additionally, it would take considerable time and manpower to extract keywords to keep up with policies as legislation and the language/terminology inevitably change. Fortunately, supervised learning algorithms should perform better in the long run as the training set grows as the data set of manually extracted keyword grows.

6. Adding terms and queries to the ontology, and building relationships also require manual work and are time consuming processes. Additions, editions, and deletions to the ontology are not quick processes and introduce a time lag between changes in policy language and its reflection in the ontology.

5.1.1 Privacy Information Framework

DOOP was constructed as a knowledge base for the vocabulary used in the privacy policies with the key goal of using it in a browser extension to help end users make informed decisions about their online privacy. The construction of this extension is out of the scope of this thesis, but a brief description is provided below to guide future research efforts.

The extension will have the following properties:

1. Collect user concerns by prompting the user to select any of the concerns that are stored in the ontology.

2. When browsing to a website with a privacy policy, automatically parse the policy, and using the keywords returned from the ontology highlight the appropriate sentences in the policy, display the policy to the user.

The extension will need to be usable, freely available, and deployed for the top web browsers to help as many people as possible.
Transparency

Previous work by (Guntamukkala et al., 2015), showed how ambiguous language used in privacy policies is making them less transparent. To encourage reading of less ambiguous sections, transparency levels of sections could be colour coded. It might also be a wake-up call to the creators of privacy policies.

Graphical User Interface (GUI)

Since this extension constantly demands user’s attention having a usable UI is going to play an important role in determining the success of the extension. The envisioned UI will need to show a dissected privacy policy where bookmarks for all of the highlighted sentences are easily accessible and properly labelled so that users can easily jump to the relevant sections without having to scroll. Other important information such as the readability score, transparency score for the policy, and colour coding for transparency should also be included in the UI.
References


Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... et al. (2001). Manifesto for agile software development.


*Terms of Service; Didn’t Read (ToS;DR).* (n.d.). Retrieved from [https://tosdr.org/index.html](https://tosdr.org/index.html) (Accessed July 12, 2016)


Appendices
Appendix A

Additional Results DOOP Validation
Figure A.1: Sentences selected for reading for query 1: ‘Does this website share my personal information with third-parties?’
Figure A.2: Sentences selected for reading for query 1: ‘Does this website share my personal information with third-parties?’
Query 2: OPP−115 Threshold 0.5

Figure A.3: Sentences selected for reading for query 2: ‘Does this website use tracking cookies?’
Figure A.4: Sentences selected for reading for query 2: ‘Does this website use tracking cookies?’
Query 3: OPP−115 Threshold 0.5

Figure A.5: Sentences selected for reading for query 3: ‘Can I opt-in/opt-out of information gathering?’
Figure A.6: Sentences selected for reading for query 3: ‘Can I opt-in/opt-out of information gathering?’
Figure A.7: Results for experiment 2 for query 2 after adding ‘cookie’ to the keywords.
Q3 with 'cookie': OPP−115 Threshold 1

Figure A.8: Results for experiment 2 for query 2 after adding ‘cookie’ to the keywords.
Appendix B

Published paper

B.1 Paper Information

**Title:** Extracting keyword and keyphrase from online privacy policies

**Keywords:** Privacy, Law, Ontologies, Taxonomy, Data privacy, Feature extraction, AlchemyAPI, keyword extraction, keyphrase extraction, online privacy policies, ontology, comprehensive taxonomy, TF-IDF, RAKE, TextRank

**Published in:** 2016 Eleventh International Conference Digital Information Management (ICDIM)

**Date of Conference:** 19-21 September 2016

**Conference Location:** Porto, Portugal

**Publisher:** IEEE

**DOI:** 10.1109/ICDIM.2016.7829792

**Electronic ISBN:** 978-1-5090-2641-8

**Print on Demand (PoD) ISBN:** 978-1-5090-2642-5
Extracting Keyword and Keyphrase From Online Privacy Policies

Dhiren A. Audich
School of Computer Science
University of Guelph
daudich@uoguelph.ca

Rozita Dara
School of Computer Science
University of Guelph
drozita@uoguelph.ca

Blair Nonnecke
School of Computer Science
University of Guelph
nonnecke@uoguelph.ca

Abstract—One of the key components of constructing an ontology is a taxonomy. Creating a comprehensive taxonomy involves extracting keywords and keyphrases from the domain corpus. It is a time-consuming endeavour that involves domain expertise and syntactic and structural knowledge of the corpus in question. In this paper we explore different keyword and keyphrase extraction algorithms for the domain of online privacy policies. To do this we used a variety of well-known techniques such as TF-IDF, RAKE, TextRank, and AlchemyAPI, benchmarked against manual annotation. We then further evaluated the performances of various algorithms over a large corpus of 631 privacy policies. Due to the inconsistent language of privacy policies algorithms evaluating single documents (RAKE, TextRank, AlchemyAPI) outperformed the one evaluating the entire corpus (TF-IDF).

I. INTRODUCTION

Websites justify collecting an individual’s information by promising a better and more user-centric web experience. However, this information is frequently shared with or sold to third parties in the pursuit of profiling users and tracking them across multiple websites [1]. Because users are often unaware of the collection and sharing of this information, and rarely knowingly consent to its data gathering, they effectively lose ‘ownership’ of their personal information and privacy [2], [3]. Privacy policies offer a glimpse into how their data is collected and disseminated, and are designed to reduce fear amongst users concerning their personal information [4]. These policies describe how users’ data are gathered, shared, and used. By law, privacy policies are often required to disclose the nature and extent of information collection [5]–[8]. But due to their legal nature, privacy policies are often hard to read [9]–[12], which is a deterrent for the average user. When a privacy policy is present, users report feeling more secure in sharing information [13]. To ease user concerns and bolster trust, companies have started introducing privacy enhancing technologies (PET) such as: opt-out mechanisms; reducing the amount of personal information collected; anonymization of personal data; and ‘layered’ policies [14], [15].

To aid users, several attempts have been made to simplify policies. The most notable of these efforts was the Platform for Privacy Preferences (P3P) [16], [17], which had limited success due to a lack of industry and developer participation. P3P also lacked proper policy validation which prevented policy developers from creating accurate policies [18]. More recently crowdsourcing (many doing little) has been used to understand privacy policies, e.g., Terms of Service:Didn’t Read (ToS:DR) [19]. The limiting factor with crowdsourcing is the large scale participation rate required to ensure success. As a result, this can lead to delays in the success of the project. To remedy this, researchers [20], [21] have tried to combine ToS:DR, natural language processing (NLP) and other machine learning techniques, with the goal of automatically inferring privacy concerns from privacy policies. Whilst these techniques work well in recognizing the pre-defined classes of privacy concerns, they still rely on quality, reliable, and up-to-date crowdsourced data which is presently lacking [20].

Another approach is to increase the transparency and understanding of privacy policies and thus provide the user with reasoning about the issues and pertinent information. One way to do this is through ontologies. Ontologies capture domain specific taxonomy and preserve context; a reasoner can then be used to reason over a particular document providing useful insights. Historically, they have been widely used in the domains of healthcare, animal and plant biology [22], but not in a legal or semi-legal domain. A key component of the ontology is the taxonomy which is constructed by extracting keywords and key phrases from the corpus of the targeted domain (privacy policies in this case). Usually domain experts are employed to aid with creating ontologies and highlighting key concepts that need to be part of the taxonomy. Recently, NLP techniques have been employed to extract keywords for the taxonomy creation [23], [24].

Key term extraction algorithms such as Rapid Automatic Keyword Extraction (RAKE) [25] and TextRank [26] have shown promising results over traditional benchmarks such the Term Frequency-Inverse Document Frequency (TF-IDF). They employ statistical learning over a custom training set by creating an array of word tokens or clusters of words, which are then ranked based on the classification of whether they are positive or negative examples of keyphrases. The classifier takes into consideration word frequency in a document, surrounding stopwords, and term clusters of words with similar meaning, as important features in a document that determine the relevance of terms. These algorithms have also been previously used to extract keywords and keyphrases from legal documents [27], [28].

To the best of our knowledge, no attempt has been made to extract keywords from privacy policies or English legal documents. However, there were attempts to extract keywords from Polish and Japanese legal documents. In the research conducted by Jungiewicz et al. [27], the Polish researchers used RAKE algorithm with a custom built stoplist of Polish words. Input consisted of 11,000 rulings from the National Appeals Chamber from the Polish Public Procurement Office. The results were qualitatively compared with results from running RAKE with a standard information retrieval stoplist. The researchers concluded that their custom stoplist worked better for the sample size that they analyzed, however, their results remains.

In a similar approach, Japanese researchers [28] proposed an algorithm using stopwords as delimiters to select candidate keywords: single keywords as well as keyphrases. The algorithm is very similar to RAKE, but uses Okapi BM25 [29] ranking function and Term Frequency - Inverse Document Frequency (TF-IDF) to calculate candidate keyword’s scores. The researchers benchmarked their algorithm against a manually annotated Japanese National Pension Act and compared their results with a similar implementation using the TextRank keyword extraction algorithm. They concluded that
their algorithm achieved an F-score of 8% higher than TextRank. Notwithstanding these efforts, keyword extraction in the realm of legal documents remains an elusive task.

One of the problems that arises in keyword extraction is that of topic relation. A common observation exploited during keyword extraction is the relatedness of keyphrases in a document. This aids in recognizing neighbourhoods from which keywords can be ranked in terms of ‘weights’ within the cluster [26], [30]. The resulting keywords becoming the desired keyword for the document. This is difficult to do in text documents where the topics constantly change (e.g., chats, personal emails, informal meetings, and personal blogs). In such texts, it is difficult to exploit the relatedness factor to select key terms for the document. This is particularly true for privacy policies as there are no standards for their writing. We found that certain privacy policies frequently omit sections related to certain topics, e.g., if a topic does not concern the website in question. Recently, Cranor et al. [1] showed through their analysis of 75 online policies that many policies do not provide enough transparency about data collection for the users to make informed privacy decisions. Using a larger dataset ensures that keywords that are extracted cover as many topics as possible.

In this paper, we investigate the performance of various state-of-the-art NLP techniques in keyword and keyphrase extraction over the online privacy policy domain. By employing different kinds of algorithms we hoped to find an algorithm that is best suited to tackle the unique features of the privacy policy text. In so doing, we wished to reduce the amount of time it takes to populate a taxonomy, and by extension construct an ontology for the online privacy domain. As the domain of online privacy evolves and laws change, we anticipate that these techniques will be transferable to the evolving privacy policies. Moreover, automating keyword extraction should reduce the cost of ontology construction, as fewer domain experts (lawyers) will need to be hired to manually do this task. Our approach is divided into two parts:

1) Compare human and algorithm performance for tackling the semantic diversity and textual ambiguity of 21 privacy policies
2) To reduce the problem of topic relation and to reinforce the results found in the first experiment, investigate how the algorithms perform over a large and diverse set of 631 privacy policies. This ensures that the results are representative of the problem domain.

The long term goal is to find the best performing algorithms and use them in a pipeline of algorithms to semi-automate the construction of a privacy policy ontology.

II. EXPERIMENT I: INITIAL BENCHMARKING

For this experiment we selected 21 online privacy policies from a variety of domains. To ensure diversity, policies were selected based on their length, transparency, comprehension (level of difficulty), intended geographic audience (U.S., E.U. Canada), industry sectors (healthcare, e-commerce, etc.), and the most visited websites1. Table I shows a domain specific breakdown. These policies were a subset of the larger dataset used in Experiment II. For keyword extraction we applied manual, TF-IDF, RAKE, TextRank, and AlchemyAPI methods.

A. Five Key Terms Extraction Methods

1) Manual Extraction: As no previous attempt has been undertaken to create a generic taxonomy for online privacy policies, there exists no “gold standard” for comparison. In order to compare the algorithms, terms were manually extracted from 21 privacy policies. The terms were carefully selected to create a comprehensive taxonomy that could be used to create a general ontology over the online privacy domain. The criteria used for term selection are described in Table II

<table>
<thead>
<tr>
<th>Domain</th>
<th>No. of websites selected</th>
</tr>
</thead>
<tbody>
<tr>
<td>Healthcare</td>
<td>1</td>
</tr>
<tr>
<td>Insurance</td>
<td>2</td>
</tr>
<tr>
<td>Banking &amp; Financial</td>
<td>5</td>
</tr>
<tr>
<td>E-Commerce</td>
<td>3</td>
</tr>
<tr>
<td>File Sharing</td>
<td>1</td>
</tr>
<tr>
<td>Search Engines</td>
<td>2</td>
</tr>
<tr>
<td>Social Networking</td>
<td>3</td>
</tr>
<tr>
<td>EU Specific</td>
<td>3</td>
</tr>
<tr>
<td>Cloud Hosting</td>
<td>1</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>21</strong></td>
</tr>
</tbody>
</table>

1) TF-IDF: TF-IDF [31] is a commonly used baseline when comparing keyword extraction algorithms. For a given phrase (j) in a patent document (k) from a corpus of size n, its term weight (wjk) is the product of its term frequency (tfjk) and the inverse document frequency (idfj).

\[ w_{jk} = tf_{jk} \times idf_j \]  
\[ idf_j = \log_2 \left( \frac{n}{df_j} \right) \]  

Essentially, it describes the importance of a term to a document whilst still taking into consideration it’s occurrence in the corpus.

3) RAKE: RAKE [25] is an unsupervised, domain-independent, language-independent, and corpus-independent method for extracting keyphrases. It is based on the observation that keywords frequently contain multiple words but rarely contain punctuation marks or stopwords. Stopwords include function words such as and, the, and of, or other words with little lexical meaning. RAKE takes a list of stopwords, a list of keyphrase delimiters, and a set of word delimiters as input, and uses them to partition the document into candidate keywords. Co-occurrences of words within these candidate keywords are meaningful and allow us identify word co-occurrence without the need for arbitrarily sized sliding windows. This way, word associations are captured in a way that adapts to text provided resulting in adaptive and fine-grained word co-occurrences that will be used to score candidate keywords. The score of each candidate keyword is the sum of scores of each of its member term co-occurrences. Word scores are based on two things: (i) its word frequency \( \text{freq}(w) \), its degree \( \text{deg}(w) \) (within the co-occurrence matrix), and the ratio of degree to frequency \( \text{deg}(w)/\text{freq}(w) \).

RAKE is also able to find key terms with stopwords within them, e.g. ‘axis of evil’, by finding pairs of candidate keywords that are adjoining and in the same order in a document. The stopword is then filled in and the score is calculated as usual. RAKE evaluates each document independently. Since, there is no standard language used by privacy policies we expected this feature to be useful in finding relevant terms.

4) TextRank: TextRank [26] is a well-known graph-based keyphrase extraction algorithm. It uses graph based ranking model
## TABLE II

**CRITERIA FOR MANUALLY EXTRACTING KEY TERMS.**

<table>
<thead>
<tr>
<th>Concept</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Legal terms</td>
<td>Online Privacy Protection Act, non-disclosure agreement</td>
</tr>
<tr>
<td>Legal organizations (government, regulatory, commercial, and computing organizations)</td>
<td>federal trade commission</td>
</tr>
<tr>
<td>Acronyms of legal organizations and acts</td>
<td>FTC, COPPA</td>
</tr>
<tr>
<td>Legal entities that can be used to define an organization or an individual</td>
<td>personal information, address, account id, internet protocol address</td>
</tr>
<tr>
<td>Data sharing</td>
<td>3rd party cookies, aggregate information, google analytics</td>
</tr>
<tr>
<td>Hosting</td>
<td>backup storage, servers</td>
</tr>
<tr>
<td>Web &amp; tech related terms</td>
<td>ad data, cookies, analytics, tracking cookies</td>
</tr>
<tr>
<td>Legal actions and legal processes</td>
<td>tracking, surveillance</td>
</tr>
<tr>
<td>Mobile privacy</td>
<td>geo-location, device identification</td>
</tr>
</tbody>
</table>

for graphs extracted from texts to rank keywords based on the co-occurrence matrix between words. Formally, let \( G = (V, E) \) be a directed graph of a set of vertices \( V \) connected by a set of edges \( E \), where \( E \) is a subset of \( V \times V \). Each vertex corresponds to a word type. A weight \( w_{ij} \) is assigned to the edge connecting two vertices, \( V_i \) and \( V_j \), and its value is the number of times the corresponding word types co-occur within a window of \( W \) words in the associated text. The score of a vertex \( V_i \) is defined by equation 3 [32]:

\[
S(V_i) = (1 - d) + d \sum_{V_j \in In(V_i)} \sum_{V_k \in Out(V_j)} w_{jk} S(V_j) \quad (3)
\]

Where \( In(V_i) \) is a set of vertices that point to a vertex \( V_i \) (predecessor), \( Out(V_i) \) is a set of vertices that the vertex \( V_i \) points to, and \( d \) is the damping factor that can be set between 0 and 1 (usually set 0.85). Scores for the vertices determine their importance, and the word types that correspond to the highest scored vertices form keyphrases. The score for \( V_i \), \( S(V_i) \), is initialized with a default value and is computed in an iterative manner until convergence.

5) AlchemyAPI: AlchemyAPI was chosen as a state-of-art industrial tool for key term extraction. It’s implementation is available via an open API\(^2\). It takes as input a text document and returns a list of significant words or phrases extracted from the document. Since it returns a set of ngram terms, we are still able to compare the results with the other algorithms.

### B. Experimental Setting

In this section, we describe the parameters used for the algorithms, pre-processing, and evaluation methods that were used in the extraction of the keywords.

1) Pre-processing: To ensure consistency of key term extraction across the corpus, we employed a series of pre-processing steps to normalize the input text. All of the input was first converted to lowercase; URLs and email addresses were removed; non-printable (as defined by the `string.printable` set in Python\(^3\)) were removed; and the remaining special characters that weren’t caught by previous filters (‘@’#, as well as other ASCII based characters from the `string.punctuation` set in Python\(^3\)) were removed. Tokenized numbers were also removed as they do not tend to add value to the taxonomy e.g. ‘1945’. We used the standard Porter Stemming Algorithm [33] from the NLTK\(^4\) library to consolidate inflected word forms to their root, e.g. ‘collection’, ‘collecting’, ‘collected’ all refer to the same concept of ‘collect’. This was done to reduce the number of variations of a term so as to not skew the validation. Furthermore, the duplicates were removed from the resulting sets.

2) Algorithm Setup: This section clarifies the configurable parameters of the algorithms that were used in the experiments.

**TF-IDF:** Scikit-learn’s `TfidfVectorizer`\(^5\) was used to implement this algorithm. TF-IDF typically operates on single words, but the Scikit tool allows `ngrams` to be extracted. To encourage the extraction of multi-word terms, the `ngram` parameter was set to \((1, 3)\) to allow the extraction of terms of length 1-3. To further improve the relevancy of candidate terms, only the top 1/5th of the scoring terms per document were selected.

**RAKE:** A well-known Python library\(^5\) of this algorithm was used in implementation for the experiment.

**TextRank** For Part-of-speech (POS) tagging, NLTK’s recommended Maxent Treebank Tagger trained on the Penn Parsed Corpora was used. We used the syntactic filters that achieved the best results recommended in the original TextRank proposal: adjectives and nouns. This was also done after consulting the results of the manual annotation. Other parameters were also chosen exactly like those that performed best in the TextRank experiments: the co-occurrence window of 2 for term relationships and undirected treatment of edges between vertices.

3) Evaluation Metrics: As it is traditionally done with information retrieval, we compared the results from each of the algorithms using 3 main scores: precision, recall, f-measure. Precision and recall are well-known evaluators of information retrieval proposed by [34]. We also calculated the Jaccard’s similarity co-efficient between the manual annotation and each of the resultant sets to understand their similarity.

**Precision:** Precision \( (P) \) is the ratio of the number of relevant terms returned from a term extraction algorithm \((\{\text{machine-selected}\})\) to the total numbers of retrieved terms from manual annotation \((\{\text{manually selected}\})\). The precision is calculated using Equation 4.

\[
P = \frac{|\{\text{manually selected}\} \cap \{\text{machine-selected}\}|}{|\{\text{machine-selected}\}|} \quad (4)
\]

**Recall:** The recall \( (R) \) of an information system is defined as the ratio of the number of relevant terms returned to the total number

---

\(^2\)http://alchemyapi.com/products/alchemylanguage/
\(^3\)http://www.alchemyapi.com/products/alchemylanguage/
\(^4\)http://scikit-learn.org/stable/
\(^5\)https://github.com/aneesha/RAKE
TABLE III
RESULTS FOR TF-IDF PARAMETERS

<table>
<thead>
<tr>
<th>Document Freq:</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Terms</td>
<td>9057</td>
<td>1520</td>
<td>907</td>
<td>645</td>
</tr>
<tr>
<td>Precision</td>
<td>4.48%</td>
<td>18.62%</td>
<td>25.47%</td>
<td>29.30%</td>
</tr>
<tr>
<td>Recall</td>
<td>49.57%</td>
<td>34.55%</td>
<td>28.21%</td>
<td>23.08%</td>
</tr>
<tr>
<td>F-score</td>
<td>8.22%</td>
<td>24.20%</td>
<td>26.77%</td>
<td>25.82%</td>
</tr>
<tr>
<td>JSC</td>
<td>95.71%</td>
<td>86.24%</td>
<td>84.55%</td>
<td>85.18%</td>
</tr>
</tbody>
</table>

TABLE IV
RESULTS FOR EXPERIMENT I

<table>
<thead>
<tr>
<th></th>
<th>TF-IDF</th>
<th>RAKE</th>
<th>TextRank</th>
<th>AlchemyAPI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Terms</td>
<td>907</td>
<td>4116</td>
<td>864</td>
<td>799</td>
</tr>
<tr>
<td>Precision</td>
<td>25.47%</td>
<td>13.17%</td>
<td>17.3%</td>
<td>17.77%</td>
</tr>
<tr>
<td>Recall</td>
<td>28.21%</td>
<td>66.18%</td>
<td>18.07%</td>
<td>17.34%</td>
</tr>
<tr>
<td>F-score</td>
<td>26.77%</td>
<td>21.97%</td>
<td>17.59%</td>
<td>17.55%</td>
</tr>
<tr>
<td>JSC</td>
<td>84.55%</td>
<td>87.66%</td>
<td>90.36%</td>
<td>90.38%</td>
</tr>
</tbody>
</table>

of relevant terms in the collection. The recall is computed using Equation 5.

\[ R = \frac{\text{\{manually selected\}} \cap \text{\{machine-selected\}}}{\text{\{manually selected\}}} \]  \hspace{1cm} (5)

F-measure: The *f-measure* or *f-score* is the harmonic mean of precision and recall [35]. The f-measure value is calculated using Equation 6.

\[ \text{F-measure} = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \]  \hspace{1cm} (6)

**Jaccard’s Similarity Co-efficient:** Jaccard index is a measure often used for comparing similarity, dissimilarity, and distance between data sets. Measuring the Jaccard similarity coefficient between two data sets is the result of dividing the number of terms that are common to both sets and the total number of terms. It is illustrated in the Equation 7.

\[ J(A, B) = \frac{|A \cap B|}{|A \cup B|} \]  \hspace{1cm} (7)

C. Results

The keyterms generated by the algorithms were compared with the manually extracted set. In order to find the best parameters to improve the scores for TF-IDF, we combined the document frequency (df) score for each term with the TF-IDF scores as demonstrated by Jacques Vergne [36]; results are shown in Table III. Due to this, we were able to achieve the highest *F-score* of 27% by keeping the minimum document frequency to 3. The parameters used for the algorithms that achieved the best *F-score* in Experiment 1 were also used in Experiment 2.

The results are summarized in Table IV. The manual annotation resulted in 829 terms being recognized as important for building a taxonomy for the online privacy domain.

III. EXPERIMENT 2: ENLARGED DATASET

In Experiment II algorithms were applied against 631 privacy policies as developed by the Data Management and Privacy Governance Lab at the University of geulph. The best performing algorithm was used as a baseline for evaluating the rest. This baseline, as created by the AlchemyAPI, was chosen by comparing the results of all the algorithms against the manual annotation from the first experiment.

A. Corpus

As with Experiment 1, described in section II, the selection criteria for privacy policies was the same. A detailed domain breakdown of the policies is shown in Table V.

B. Results

The experimental setup of this experiment was same as the first experiment for all four algorithms. Instead of choosing TF-IDF as the baseline as is generally done, we evaluated the results from all of the algorithms against the results from the manual annotation. AlchemyAPI had the highest *F-score* of 4.85%, as such it was chosen as the baseline to evaluate the results from the rest of the algorithms. The results are summarized in Table VI.

IV. DISCUSSION

Results from Experiment 1 show no significant difference between TextRank and AlchemyAPI’s performance when compared to manual extraction. TF-IDF performed the best with an *F-score* of 27% and RAKE was a close second with 22%. However, in Experiment II using the larger dataset AlchemyAPI performed the best with RAKE a close second 25%.

From Table III, it is clear that the most important keywords appeared at least thrice in the same policy. It was unexpected however for TF-IDF to perform as well as it did, since the language used in all of the policies were quite different; some were legalese (healthcare, financial institutions), and others easier to read (Dropbox, Google). This diversity is important because TF-IDF scores each term not only by how it appears in one document but also its scarcity of appearance elsewhere in the corpus. As the language across the corpus was inconsistent, we expected TF-IDF to perform the poorest.

On the other hand, RAKE, TextRank, and AlchemyAPI only consider a single document at a time when extracting terms instead of the
Our research forms the first step in creating a context aware system for real-time privacy policy evaluation.

For this reason, it was expected that evaluating a larger dataset of privacy policies would eliminate, or at least reduce, the gaps in the individual algorithm performances; it did not. One reason for this could be that since the AlchemyAPI algorithm was used as a benchmark, it could be negatively affecting the scores in the second experiment as calculating scores based on it could bias the results in favour of algorithms that evaluate each document separately. In contrast, keyword extraction done by domain experts would likely be more neutral. For the future, we anticipate that increased manual annotation will provide a better baseline to score the algorithms against.

Our results were affected by four major types of errors: overgeneration, redundancy, infrequency, and evaluation errors as described by Hasan and Ng [37].

**Overgeneration errors** are a type of precision errors that occur when the algorithm incorrectly chooses a candidate term because it contains a word that appears frequently in the document or corpus, but at the same time outputs similar but incorrect candidate terms because they too contain a frequently occurring word. This contributes to a high degree of divergence within the results for that algorithm. A prime example of this are the results from the TF-IDF which performed very poorly because it reported *unigrams, bigrams* and *trigrams* that were often subsets of each other resulting in a very high number of terms being generated. RAKE and TextRank did not suffer much from this problem because both calculate scores for candidate keyphrases based on frequency of groups of constituent words appearing in that particular order whereas TF-IDF treats each of its constituent word as a separate word, in order to calculate its score, and sums individual word scores to calculate the candidate keyphrase score. An example of this is illustrated in Fig 1.

**Redundancy errors** are a type of precision errors that occur when an algorithm correctly identifies a candidate term or phrase, whilst simultaneously selecting a similar candidate term or phrase that are semantically the same. For example, ‘account use’ and ‘account usage’ refer to the same concept of anonymous usage, but are reported as separate keywords. All the algorithms were affected by this error. However, TextRank had fewer errors overall because of the hard limit of the co-occurrence window to 2 words per keyphrase.

**Infrequency errors** are a type of recall error that occur when an algorithm doesn’t recognize a correct candidate keyphrase because of its scarcity in a document. It is challenging to address this type of error because algorithms rarely choose candidate terms that only appear once or twice in a document. Furthermore, for TF-IDF we had forced all candidate terms to appear at least thrice in a document to generate more relevant results.

**Evaluation errors** are a type of recall error that occurs when the scoring algorithm incorrectly scores a keyphrase as not a keyword despite it being syntactically similar to a gold keyphrase. Part of this type of error was dealt with by scoring the stemmed output of the keyphrases. Still there was not a concrete way of eliminating all of the errors. For example, ‘account use’ and ‘account usage’ should be recognized as the same concept but is calculated as separate keyphrases.

It is important to remember that these results are preliminary and only reflect results of 4 well known algorithms. The experiments were primarily conducted to see how effective keyword extraction algorithms are over a semi-legal and legal domain such as online privacy. Further research will entail running more algorithms over a larger corpus. We expect that increasing the number of policies will increase the diversity of the privacy policies by enough that terms currently viewed as unimportant due to the low occurrence in the corpus might be scored higher for TF-IDF. Further research will involve implementing supervised learning algorithms such as Kea++ [38] using our manually extracted terms as a training set. Topic-based clustering such as CommunityClustering [39] may prove promising. Finally, with an extensive and comprehensive taxonomy we intend to construct ontologies that cover the entire domain of online privacy policies.

V. CONCLUSION

In this paper, we presented a systematic study of keyword and keyphrase extraction over the domain of online privacy policies. We tested 4 prominent automatic keyword extraction algorithms against manual annotation and found that TF-IDF performs well against manual annotations on a small dataset of 21 policies, and AlchemyAPI performs marginally better than TF-IDF on larger datasets.

Due to the lack of standardized language used in online privacy policies, algorithms that evaluated individual documents (TextRank, RAKE, AlchemyAPI) outperformed by a significant margin algorithms that extracted keywords from the larger corpus (TF-IDF). Our results confirm that using natural language processing techniques for keyword and keyphrase retrieval from legal and semi-legal documents remains a challenging task.

Our preliminary results will guide further research in the field of online privacy policies and machine learning. For these experiments we used the entire dataset of heterogenous privacy policies; it will be worthwhile to evaluate the performance of algorithms against datasets for specific privacy policy domains e.g. cloud service providers, e-commerce, and social networking. Similarly, algorithm performance could be assessed for policies that are divided into common sections like share, collect, consent, and store. Finally, we intend to use more content-based single document algorithms and explore supervised learning algorithms over larger data sets of online privacy policies to help construct a online privacy ontology. In addition, it will be useful to know how trained keyword extractors perform against the algorithms and against domain experts (privacy researchers and lawyers). Our research forms the first step in creating a context aware system for real-time privacy policy evaluation.