Find Conflicting Privacy Policies on a Webpage

Fangkai Wang

Master of Science

Computer Science

School of Informatics

University of Edinburgh

2016

Abstract

When a website loads content from third-party websites, user's information is sent to third-party websites as well. Both the first and third-party websites have their own privacy policies. There is no guarantee that the privacy policy of first and third-party match each other. We have built a privacy policy gather to collect privacy policies from both first and third-party websites, compared the topic similarity between policies and measured the readability and cost of time to read privacy policies.

Acknowledgements

I would like to extend my sincerest gratitude to my supervisor, Dr Kami Vaniea, for her patient guidance, advice and encouragement throughout all stages in my project. I have been extremely lucky to have a supervisor who cares for my work so much, and who responds to my questions and queries so promptly. I also want to express my thanks to Prof. Bonnie Webber who has provided valuable advice and insight.

Thank you also to my parents, friends for your continually supports and encouragement during these days.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Fangkai Wang*)

Table of Contents

Chapter 1

Introduction

It is common for modern websites to use services provided by third-party websites. For example, there is an advertisement image on `yahoo.com` which is provided by google. User's information collected by the first-party websites that user originally visits is also transported to third-party websites for any other uses when using third-party websites' services. Both the first-party website that user initially visits and third-party websites have their own privacy policies which state the collection and use of the user information(United States Government Accountability Office, 2013).

However, there is no guarantee that these privacy policies match each other. For example, assume a user visits website A which loads a image from B, user's information is sent to B as well. A's privacy policy states that A will not **sell** user's data while B claims that B will do so. This potential conflicts violate A's privacy policy and can result in disclosure of user information collected by A.

The situation becomes worse with behavioral advertising industry which has grown dramatically in recent years (Hoofnagle et al., 2012). Behavioral advertising using several techniques to collect user's information, directly observable (e.g., IP address) or indirectly observable information (e.g., fonts enabled on browser), to build profile in order to track users across websites and provide better customized advertisements (Nikiforakis et al., 2013). The user's profiles generated by user initially visiting websites, the first-party websites, is also transferred to third-party websites contacted by the first-party websites to provide behavioral advertising (United States Government Accountability Office, 2013). Such data sharing for advertisement is totally fine when we only concern about single website since their privacy policy clearly states they will or will not share user's information for advertising. However, things change when the first-party websites interacting with third-party websites. Though either first-party websites or

third-party websites have their own statements about advertisement in their privacy policies, they may treat user's information differently. Even user visits another websites, third-party websites are able to use such information to track user, provide tailored advertisements or any other uses. For example, share these information with more third-party websites so that user's information is spread widely on the web (United States Government Accountability Office, 2013). Figure 1.1 demonstrates this data transfer when a user visits several first-party websites.

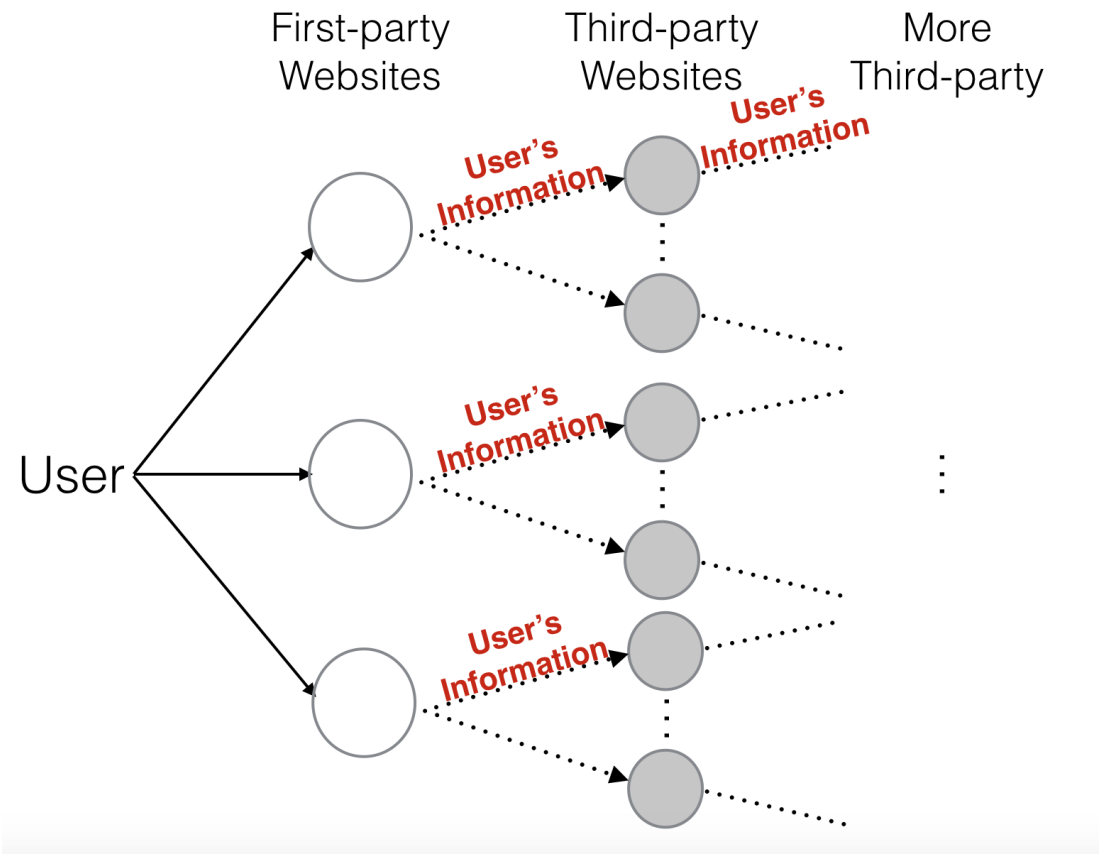


Figure 1.1: User's Information Transfer between First and Third-party Websites

Although many privacy laws have been introduced to request website and app companies to include a privacy policy which states its data collection and use [1], current law framework does not fully reflect such changes in technology and marketplace (United States Government Accountability Office, 2013).

Therefore, we try to bring some insights into this issue– when a websites using service provided by third-party websites, identify the exists of potential conflicts statements

[1]For example, EU: Data Protection Directive 95/46/EC (The European Parliament and The Council of The European Union, 1995), U.S.: Children's Online Privacy Protection Rule (Federal Trade Commission, 2013)

between the privacy policies of first and third-party websites.

To achieve this goal, we have developed a privacy policy gather including a third-party links finder and a privacy policy extractor (PPE), then we compared the privacy policies via NLP and measured its usability i.e., readability and cost of time to read:

1. **Privacy Policy Gather (PPG):** Given a target websites, PPG aims to collect the privacy policies of both the first-party website and connecting third-party websites. PPG first loads a first-party website in a real browser in order to execute all the Javascripts. Then PPG finds all the connecting third-party websites via third-party link finder which generates a list of URLs of connecting third-party websites. Finally, PPG uses PPE to extract the privacy policy of all the URLs including first-party website and store results on database.

   (a) **Third-party Link Finder (TLF):** A TLF is designed to find the URL of third-party websites. Our TLF achieve this goal via capturing network traffic when PPG loading the first-party websites. All the requesting URLs can be regarded as "third-party" at this stage. These URLs definitely contain the URL of first-party websites but we will filter them out in our data pre-processing phase.

   (b) **Privacy Policy Extractor (PPE):** A PPE focus on finding the privacy policy pages and extracting the text of privacy policy. PPE first visits the domain of given URL. For example, given `http://example.com/flower.jpg`, PPE visits `http://example.com`. Then it extract the privacy policy link on the page by its simple text matching (e.g., match for *privacy*, *privacy policy*). The matching could fail since some requesting URL is designed to provide services rather than for human to read. PPE will query WHOIS server of failed URL to find the URL of its registrant and extract registrant's privacy policy same as previous steps as the third-party privacy policy of failed third-party websites.

2. **Comparing Privacy Policies** We compares the privacy policy of each third-party websites to the first-party website using several metrics to find potentially conflicting privacy statements using techniques from natural language processing (NLP). This is the most challenging part of the project. It is straightforward that if we can parse the privacy policy into machine readable format and find exact negation.

Unfortunately, this is not a viable solution. CMU has a large project on parsing privacy policies [2]. We consulted with them and determined that accurately converting a privacy policy to a machine readable format is unfeasible. But they suggest that using topic modeling, which produce several topic words for the given text, on paragraph level is possible to compare the topic similarity. We also had a discussion with a NLP expert from the faculty who is doing research on negation, she indicated that finding accurate negation in the privacy policy is also unfeasible. However, she suggested that counting the frequency of sensitive words (e.g., share, disclose) can be a reasonable start point. The reason is that we seemed more likely to repeat a thing many times if we would do that (positive) while only a few times if we would not do so (negative).

Based on their suggestions, we decide to shift to measure the topic similarity of privacy policies. We have counted sensitive words and applied topic modeling on our collected privacy policies.

Another question we are interested in is the usability of privacy policy, in particular, the cost of time to read privacy policies as well as their readability. Many studies reported that privacy policy is difficult to understand ((Hochhauser, 2001), (Graber et al., 2002), (Luger et al., 2013)). McDonald and Cranor (2008) reported that it will cost 181 hours per year for individual to read all the privacy policies of the websites he visited in a year and 39.9 billion hours per year for all U.S. Internet users in 2008. These numbers implies that reading is a considerably time-consuming task: need much time to comprehend. Luger et al. (2013) argued that the most complicated privacy policy is as complex as The Prince in their case study on UK Energy's websites.

To further extend above studies, we want to reproduce these results on world top rank websites with the presence of implicitly user information exchange between first and third-party websites.

We have collected privacy policies of top 100 U.S. websites ranked by `alexa.com`[3] as well as the privacy policies of the third-party websites each top sites connecting via our PPG. The collection contains 192 distinct privacy policies (79 from first and 113 from third-party websites). We found following results:

1. **Sensitives Counting**: We found this argument is not true. For the most sensitive word *share*, 31% privacy policies that "share" occurs less than once (including

[2]The Usable Privacy Policy Project, `https://usableprivacy.org`

[3]The top 500 sites on the web, http://www.alexa.com/topsites/countries/US, data retrieved on 30 June 2016

once) “will not share” user’s personal information.

2. **Topic Similarity**: We developed a metric (topic similarity score) and corresponding algorithm to measure the topic similarity between first and third-party websites’ privacy policies. By applying this algorithm to the collected privacy policies, we found the frequency of topic similarity score distributed similarly to normal distribution and topic similarity varied from extremely dissimilar to considerably similar. This implies that further investigation can be done on those extremely dissimilar policies to identify exact conflicts and violation.

3. **Usability of Privacy Policy**: From the result of measuring readability, we found 85.7% of privacy policies require U.S. high school education level to read and 51.5% require senior U.S. high school level (12th grade). For the cost of time to read, we found a user need to read at least 1.92 privacy polices (including third-party) per first-party website he initially visits and it will cost him 487 hours a year and 83 billion hours for the U.S. nation.

The remaining chapters are structured as follows: Chapter 2 introduces essential background. Chapter 3 details how we design and implement a semi-automatic PPG and collect privacy policies on Alex top 100 U.S. websites. Chapter 4 presents our analysis on the data set including comparing topic similarity and measuring usability. Chapter 5 demonstrates the result and our discussion. Chapter 6 summarizes the report and identify future work.

Chapter 2

Background

In this chapter, we will introduce essential background knowledge that helps reader better understand our project.

2.1 Privacy Policy

A privacy policy is a legal document that states how a website collects and use their users' information. Many privacy laws enforce websites to include a privacy policy. There is not a general answer of what information should be included in a privacy policy, but it is common to see the following information in a website's privacy policy:

- What information is collected?
- How the collected information will be used?
- User's choice about what/how his information is collected (e.g., opt-out, the websites will not collect this user's information anymore).
- Whether/how website share user's information
- Security of collected information
- How the updates on this privacy policy will be communicated

Figure 2.1 illustrates the privacy policy from `google.com`[1] which clearly demonstrates above elements.

[1] `https://www.google.com/intl/en/policies/privacy/?fg=1`, data retrieved on 4 August 2016

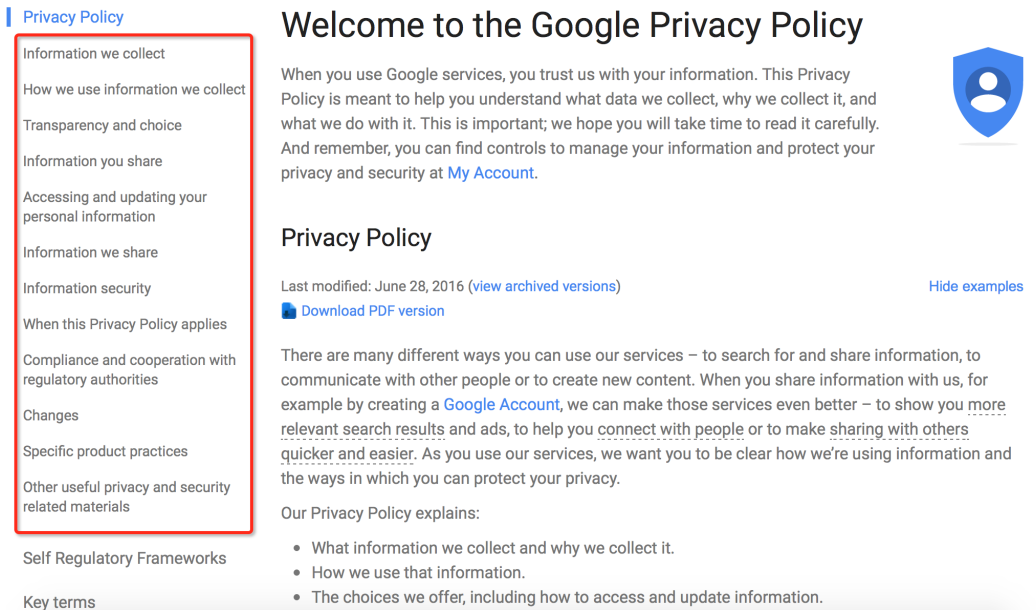


Figure 2.1: Part of `goole.com`'s privacy policy. The titles of the remaining sections are hightlighted in red rectangle.

More specific privacy statements depend on the governing privacy laws. For example, according to Childrens Online Privacy Protection Act (COPPA), if a website also provides services for children under age 13, its privacy policy should clearly describe their information practice for the personal information they collect from children (Federal Trade Commission, 2013).

2.2 Natural Language Processing Essentials and Topic Modeling

We will first introduce some basic natural language processing (NLP) terms before further explaining topic modeling. In the field of NLP, a *corpus* is a collection of documents. Each document is a collection of *terms* which is the smallest language unit we interested in (e.g., an English word). Therefore, a document can be represented as a n-dimensional vector. Each dimension represents a term occurs in the document and the value is a function of number of times that term occurs in the document (e.g., frequency). In this way, the corpus can be represented as a *term-document* . Rows are term frequency vector for each document and columns are terms in the corpus.

Consider a simple corpus which only contains 3 documents and only one sentences in each documents.

```
doc1={"hello world hello world"}
doc2={"this is the second document"}
doc3={"this document has one sentence"}
```

By counting each the word frequency of each document, we can obtain the corpus in the form of term-document matrix as follows (Table 2.1):

Table 2.1: Term-document matrix of example corpus

	hello	world	this	is	the	second	document	has	one	sentence
doc1	2	2	0	0	0	0	0	0	0	0
doc2	0	0	1	1	1	1	1	0	0	0
doc3	0	0	1	0	0	0	1	1	1	1

This representation is also known as *bag-of-words* model (BOW) that is only concerned about the frequency (or other functions of the number of times a term occurs in a document) of each term in a document and ignores the order of words.

Now we have converted the corpus from strings to term-document matrix, we can further transform this matrix by topic modeling. Topic modeling is a set of algorithms for discovering themes (topics) in documents (Blei, 2012). Given a document in the form of term frequency, the algorithm generates topic words from the document. Topic words offer us a general sense of what this document is talking about and can help us build connections between documents based on their topics (Blei, 2012).

The first topic model, latent semantic indexing (LSI), was introduced in 1998 by Papadimitriou et al.. It utilized singular value decomposition (SVD) to find the topics (Papadimitriou et al., 1998). Later, Hofmann developed probabilistic LSI which provided a proper generative data model and fitted model to data to find topics. (Hofmann, 1999). Latent Dirichlet Allocation (LDA), developed by Blei et al., extended pLSI and overcame the overfitting issue with pLSI by assigning a Dirichlet prior to the distribution of topics (Blei et al., 2003).

From the perspective of LDA, a document (in the form of BOW)is a distribution of several topics. Each topic has its own contribution to form the document. A topic is a distribution of words (Blei et al., 2003). To demonstrate how LDA works, assume we have an article (100 words) from pet magazine and want to rebuild this article. From

a LDA point of view, this article is a mixture of *two* topics: 30% from *cat* and 70% from *dog*. Topic *cat* has 3 words: 0.2 quiet, 0.5 alert, 0.3 climb and topic *dog* has 3 words as well: 0.3 bark, 0.4 bone, 0.3 fetch. The numbers is the probability of the words. Now we can rebuild the article word by word. For each word, we randomly choose a topic (e.g., dog) then randomly choose a word under that topic as output (e.g., bark), repeat until we generate all words. LDA reverse above steps, it discovers most probable distribution of topic for the give document and most probable distribution of words for each topics.

2.3 Usability of Privacy Policy

Many studies argued that reading online privacy policies was a tough task for normal Internet user: intensive, difficult to understand and cost much time to read (Jensen and Potts (2004), McDonald and Cranor (2008), McDonald et al. (2009), Luger et al. (2013)). We will briefly explore these arguments.

2.3.1 Readability

The issue that online privacy policies are difficult to read has been found on different categories of websites ((Hochhauser, 2001), Graber et al. (2002), Luger et al. (2013)). (Hochhauser, 2001) reported that 60 financial privacy notices required third to fourth college reading level which was behind the junior high school reading that the privacy law (Hochhauser, 2001). Graber et al. analyzed the reading level of 80 health websites using three readability level: using the Flesch[2], the Fry[3], and the SMOG[4]. The average readability level required second year college education level to understand (Graber et al., 2002). Luger et al. also discovered that the most complicated privacy policy from UK Energy are as complex as The Prince and the longest one even had more than 7000 words (Luger et al., 2013).

Aforementioned readability metrics (Flesch, Fry and SMOG) are formulas that calculate the readability of text as U.S. education level based on several factor of the text (e.g., numbers of syllable word, sentences). Table 2.2 illustrates some readability metrics of the privacy policies of `google.com`, `facebook.com` and `amazon.com`:

[2]Flesch-Kincaid Grade Level
[3]Fry Readability Formula
[4]Simple Measure Of Gobbledygook

Table 2.2: Readability metrics of google.com, facebook.com and amazon.com

Website	Flesch-Kincaid Grade level	Gunning Fog Score	SMOG Index
google.com	11.4	14	10.5
facebook.com	11.2	13.5	10.8
amazon.com	11	12.5	10.5

By using these metrics, we can quickly understand which (U.S.) education level is required to read the given text. For example, level 14 of `google.com` using Gunning Fog Score means that it required second year college education level.

2.3.2 Cost of Time to Read

In 2008, McDonald and Cranor studied the cost of time to read privacy policy. They measured the time to skim and time to read entire privacy policy for individual and the nation. They reported that if one tries to read the privacy policies of all the websites he visits word by word, it will cost him 181 hours per year and 53 million hours for the nation (McDonald and Cranor, 2008). The results indicates that reading privacy policies is a time-consuming task so that people may prefer to ignore the privacy policy.

This study also motivates us to ask the following question: assume a user knows there are many third-party websites are connecting with site he initially visits, how much time will it cost to read all these privacy policies? We will present our results on Section 4.2.

Chapter 3

Data Collection: Privacy Policy Gather (PPG)

Gathering privacy policies automatically across many different website is not trivial. Even to find the privacy policy of one website is not trivial as well (Liu et al., 2014). Though many well-regulated websites have a "privacy" link on their main page, the text of that link could vary (e.g., privacy, privacy policy). Even the URL of privacy policy page is identified, extracting the text of privacy policy is also challenging since each website has its own pattern of placing its HTML documents. For example, the privacy policy page could be distributed over several pages, buried deep within the website, or mingled with "Terms of Services" and the format could vary as well (e.g., in PDF, HTML, etc.) (Liu et al., 2014). Since we will also need to preserve the structure of privacy policy as many as possible to compare topic similarity on paragraphs level, fully automatic PPG is not feasible. We therefore used a semi-automatic approach for this problem. Given a set of first-party websites, we first visit each first-party website and find the contacting third-party websites, then visit each websites and try to find the actual URL of privacy policies as possible as we can. The previous processes are done by machine. Finally we manually examine the found URL, correct the URL if necessary and extract the text content of privacy policies.

3.1 Design Requirements

Given the URL of a first-party website, the task of PPG is to extract the privacy policy of first-party website and its contacting third-party websites. Therefore, we identify following requirements:

1. **Real browser**: Using real browser to load the websites in order to execute Javascript so the first-party websites can communicate with third-party websites.
2. **Record requests**: Be able to record the URL of each third-party websites. It should record all the resource requests when loading first-party websites.
3. **Best effort extracting**: For each main and third-party websites, make best effort to find the actual links of privacy policies in order to alleviate burden for further manually extracting privacy policy as much as possible.
4. **Data persistence**: Store results in database.
5. **Multithreading**: There will be a large number of websites to visit and extract privacy policies from so the collecting speed is also a concern.

3.2 Detailed Design

Based on the requirements above, we designed the PPG as follows:

3.2.1 Real Browser

Using Selenium webdriver to simulate real user browsing websites. Selenium is a browser automation tools, it can launch various browsers (e.g., Firefox, Chrome, etc.) and provide rich APIs to manipulate HTML elements on the web page (e.g., click a button on the web page by *element.click()* function) and configure the browser as well[1]. In addition, we prefer to use Firefox as our browser since Selenium provides better support for Firefox than other browsers.

3.2.2 Record Requests

When loading the first-party website in a browser, monitoring the network traffic generated by that browser can capture the URL of each contacting third-party websites. This can be accomplished by Firefox extensions, namely, Firebug and HarExportTrigger. The former can capture the network traffic generated by the browser when visiting a website and the latter can export the traffic into Http Archive (HAR) file. HAR is a JSON based format developed by World Wide Web Consortium (W3C) and are designed to be used by browser to export the performance data including network

[1] Selenium. `http://www.seleniumhq.org/`

traffic when loading a web page. Figure 3.1 demonstrates an example entry in HAR file generated by Firebug when visiting `bbc.co.uk`. We can parse the HAR file and

Figure 3.1: Example entry in HAR file generated by Firebug when visiting bbc.co.uk. The URL of the requesting resources and its domain name is highlighted in red rectangle.

extract the URL of third-party websites for each first-party websites.

3.2.3 Best Effort Extracting

The third-party websites recorded by capturing network traffic can not be visited directly since not all of these websites is designed for human to visit. Many of them are created for exchanging data between servers and have no visible pages for human to read. Figure 3.2-3.3 illustrates this issue. Figure 3.2 presents the network traffic captured by Firefox Web Developer tool when visiting `yahoo.com`: We can see a request is sent to `http://pagead2.googlesyndication.com/` which can be thought of as a third-party website when we initially visiting `yahoo.com`. While we directly visit this website, we will see a 404 error indicating that there is not a main web page for us and privacy policy pages as well (see Figure 3.3).

We therefore argue that replacing this human inaccessible third-party websites with its domain name registrant's website and privacy policy as well. These can be achieved by querying *WHOIS* server of the requesting websites. WHOIS service provides public access to the directory that hosts the contact and technical information of the domain name holder (registrant). These information is accurate and accessible since registrants

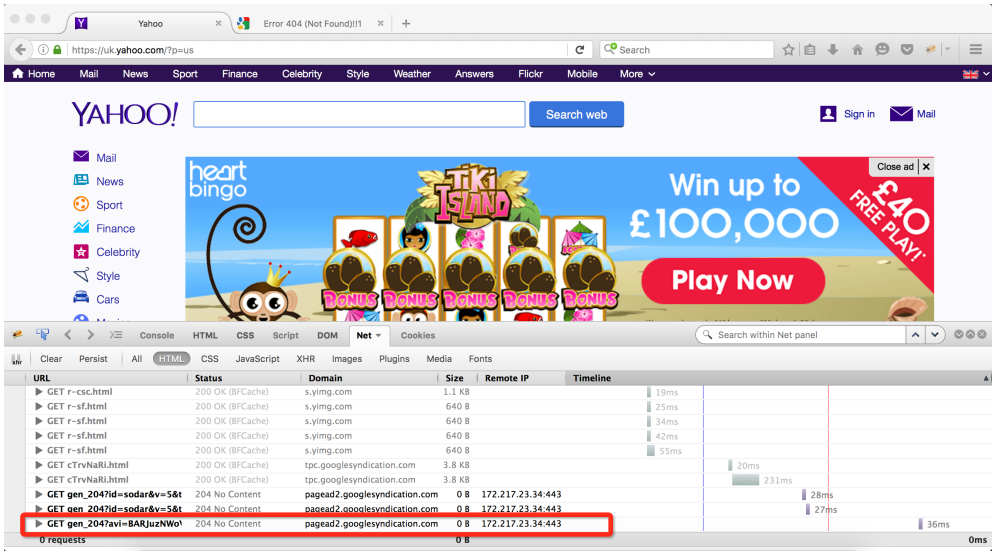


Figure 3.2: Network traffic when loading yahoo.com. An identical request sent to third-party website is highlighted in red rectangle.

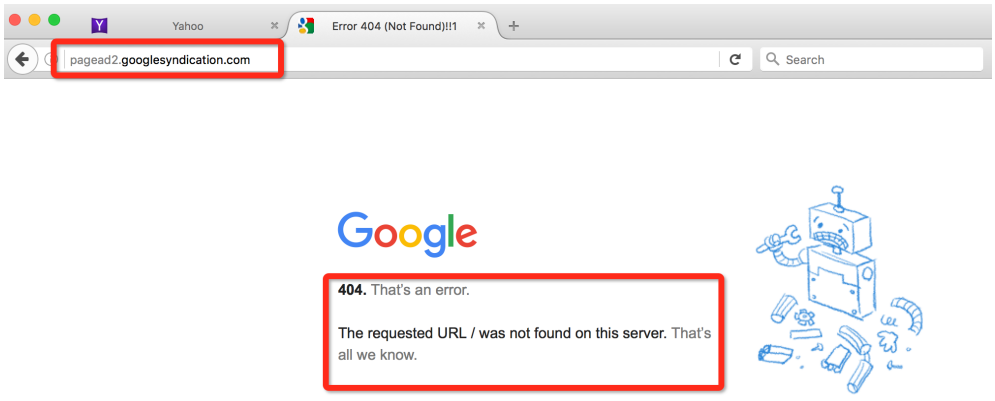


Figure 3.3: 404 error when visiting http://pagead2.googlesyndication.com/. This websites is not created for human.

are obligated to provide these information when registering a domain name[2].

In general, we first visit the URL produced by capturing network traffic then attempt to find the link of privacy policy page by matching the text of the link elements (e.g., "privacy", "privacy policy"). We would not expect more complicated matching logic to find such links since full automation extracting link of privacy policy is not feasible (Liu et al., 2014). The privacy policy page could be buried deep within the website, mingle with "Terms of Use" and in different format (e.g., PDF, HTML, etc). Instead, we do our best to identify the link of privacy policy page of a website automatically, leaving the failed website for later manual processing. If we are not able to match any link, which indicates that this website is not designed for human to read or the privacy policy pages are buried deep in the website, we will query WHOIS server of the failed websites and again try to match the text of link. This is where the argument that "Best effort" comes from, simple text matching logic along with querying WHOIS, to alleviate the burden of later manual extracting text of privacy policy.

3.2.4 Data Persistence

In order to separate data access layer from business logic layer, we will use Mybatis[3] as our data persistence framework. Mybatis alleviates the burden to write JDBC codes, we can write our database connecting configuration and SQL query sentence in XML and access the database via Plain Old Java Object.

3.2.5 Multithreading

We will use WebCollector to achieve multithreading. WebCollector is a Java-based web crawling framework and convenient to create a multi-threading web crawler[4]. Though gather several privacy policies across many websites is not an identical web crawling task, we can still utilize web crawling framework to to manage our threads.

3.3 Implementation

The privacy policy gather (PPG) comprise three components: driver program, third-party link finder (TLF) and privacy policy extractor (PPE). Given the URL of first-

[2]https://whois.icann.org/en/primer, retrieved on 8 August 2016
[3]http://www.mybatis.org/mybatis-3/
[4]https://github.com/CrawlScript/WebCollector

party websites, PPG finds the URL of the privacy policies of first-party websites and its contacting third-party websites. Then we extract the text of each privacy policy by visiting each URL of privacy policy manually (some URL may not be the desired privacy policy page so need to be reviewed). Figure 3.4 demonstrates the architecture of PPG.

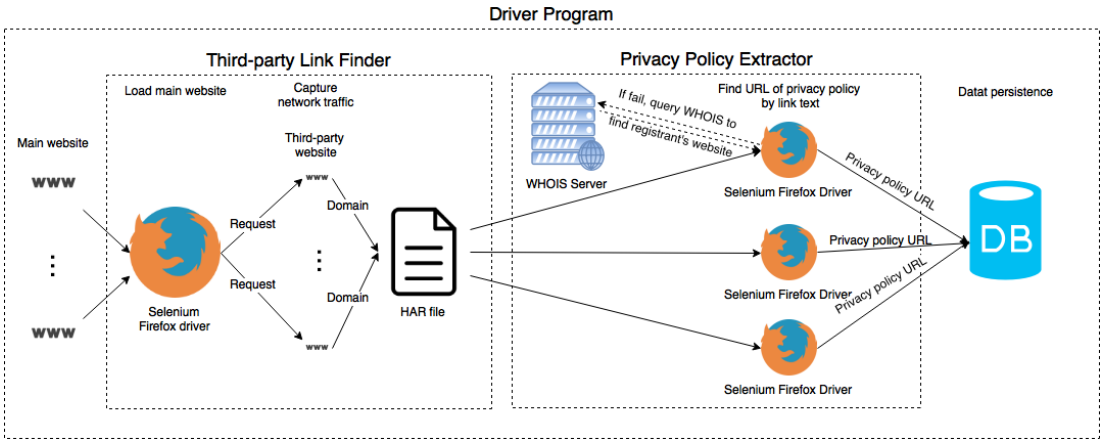


Figure 3.4: Architecture of PPG. PPG consists of three components: driver program, third-party link finder and privacy policy extractor.

The following subsections describe detail implementation of each component.

3.3.1 Driver Program (DP)

Driver program takes responsibility of the following:

1. Reads the URL of first-party websites from file.

2. Interacting with TLF and PPE. It first feed TLF with the URL of one first-party website then reads the list of third-party URL generated by TLF and send each of them to a PPE to extract the URL of privacy policy pages. Each PPE runs on single thread with a dedicated Firefox. Finally DP format the result each PPE and store them in database.

3.3.2 Third-party Link Finder (TLF)

Given the URL of a first-party website, TLF outputs the URL of contacting third-party websites when loading first-party website in specific time period. Figure 3.5 shows the work flow of TLF.

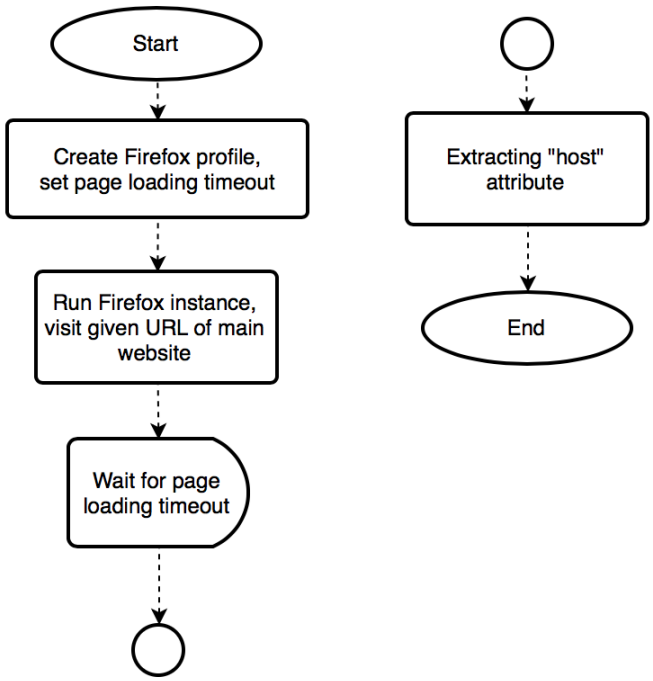


Figure 3.5: Work flow of TLF.

It first creates the Firefox profile to configure Firebug and HarExportTrigger including enabling these two extensions and set amount of time to wait before exporting the HAR file (page loading timeout). Then TLF starts a Firefox instance using Selenium with cookies and browser cache being cleaned to ensure that every visit to a first-party website is in the same state, visits the first-party website and sleeps until HAR file is generated by HarExportTrigger. All the network traffic generated by visiting the given first-party website are recorded by Firebug and exported to HAR file. TLF will parse the corresponding HAR file and extract the host of requesting resource as the output URL. Parsing is straightforward since HAR is a JSON-based format (see Figure 3.6) We can easily access the “host” attribute in the each HAR entry as the output of TLF. Finally TLF terminates the Firefox in order to save computing resources and outputs the URL list of third-party websites.

3.3.3 Privacy Policy Extractor (PPE)

PPE focus on extracting the URL privacy policy page for the given website. As shown in Figure 3.7, PPE first run Firefox instance via Selenium with cookies and browser cache being cleaned for consistence. Then attempts to identify the link of privacy policy page by simple text matching, specifically, identify a HTML link ($< a >$)

Figure 3.6: Example entry in HAR file generated by Firebug when visiting bbc.co.uk. The URL of the requesting resources and its domain name is highlighted in red rectangle.

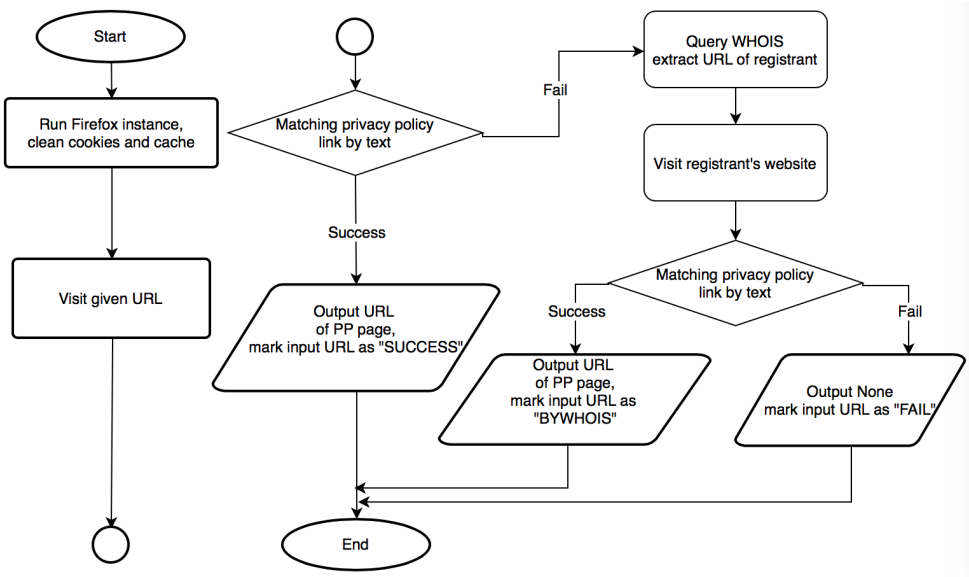


Figure 3.7: Work flow of PPE. PP refer to “privacy policy” in the figure. PPE attempts to find the URL of privacy policy page as possible as it can by trivially matching link text and querying WHOIS.

element whose “text” attribute is equal to “privacy”, “privacy policy”, etc. Selenium provides rich APIs to locate an HTML element in a webpage. For example, the following code snippet demonstrates how to find a link element by its text[5].

```
// <a href="http://www.google.com/search?q=cheese">cheese</a>>
WebElement cheese = driver.findElement(By.linkText("cheese"));
```

If matching fail, implying that the visiting website is not designed for human to read or the link of privacy policy pages is complicate for machine to locate, PPE will query the WHOIS server of the visiting websites to find the website of the its registrant as alternative. Then PPE searches for link of privacy policy page on the registrant’s website by the same simple text matching. If matching fail again, then mark the input URL as *FAIL*, otherwise mark as *BYWHOIS* and output the URL of privacy policy page. Table 3.1 illustrates three potential outputs of PPE for a given input URL and explanation and Figure 3.8 demonstrates corresponding examples for each output when extracting privacy policy page of third-party websites of `yahoo.com`.

Table 3.1: Potential outputs of PPE for the given URL

id	Status	Output URL	Explanation
1	SUCCESS	URL of privacy policy page	the URL is extracted from original third-party website
2	BYWHOIS	URL of privacy policy page	the URL is extracted from registrant’s websites
3	BYWHOIS	None	querying WHOIS success but text matching on registrant’s websites fail
4	FAIL	None	querying WHOIS fail

In the following, we will present details about implementation of querying WHOIS. According to RFC 3912, the WHOIS server listens on TCP port 43[6] so we need to send our query to port 43 of the hosting server. Figure 3.9 illustrates a sample querying procedure from WHOIS protocol specification[6].

Figure 3.10 presents an example WHOIS report for querying `http://ssl.gstatic.com` on WHOIS server `whois.markmonitor.com`. We can using regular expression to

[5]Example from Selenium documentation, `http://www.seleniumhq.org/docs/03_webdriver.jsp`, retrieved on 10 August 2016

[6]RFC 3912, `https://tools.ietf.org/html/rfc3912`, retrieved on 10 August 2016

id	Status	Third-party	Policy URL	ifByWhois	Registrant's websits
1	SUCCESS	https://pm.w55c.net	https://www.dataxu.com/about-us/privacy/	0	
2	BYWHOIS(SUCEES)	http://mpp.vindicosuite.com	http://vindico.com/privacy-policy.php	1	vindicogroup.com
3	BYWHOIS(FAIL)	https://pixel.rubiconproject.com		0	
4	FAIL	https://rp.gwallet.com		1	radiumone.com

Figure 3.8: Example output of PPE when extracting privacy policy page of of yahoo.com's third-party websites. Status bit that "ifByWhois=1" indicates this "policyURL" is extracted from registrant's websites otherwise from orginal third-party websites.

```
client                                  server at whois.nic.mil

open TCP    ---- (SYN) ------------------------------>
            <---- (SYN+ACK) -------------------------
send query  ---- "Smith<CR><LF>" --------------------->
get answer  <---- "Info about Smith<CR><LF>" ---------
            <---- "More info about Smith<CR><LF>" ----
close       <---- (FIN) ------------------------------
            ----- (FIN) ------------------------------>
```

Figure 3.9: Example WHOIS query procedure. Client search for WHOIS information of "Smith" on the host `whois.nic.mil`.

extract registrant's website from "Registrant email" entry. However, there are three problems with WHOIS:

1. **Query Format and WHOIS Server**: As specified in RFC 1035[7], a domain name is formatted as follows:

   www.example.com

   - **Hostname**: www
   - **Domain**: example
   - **Top Level Domain (TLD)**: com

   In practice, the WHOIS server only supports query formatted in "domain+TLD" so the "Hostname" need to be filtered out. We can separate each part of domain by character '.' and keep last two parts. However, some domain names also

[7]RFC 1035, Domain names-Implementation and Specification. https://tools.ietf.org/html/rfc1035, retrieved on 12 August 2016

```
Registrant Name: DNS Admin
Registrant Organization: Google Inc.
Registrant Street: 1600 Amphitheatre Parkway
Registrant City: Mountain View
Registrant State/Province: CA
Registrant Postal Code: 94043
Registrant Country: US
Registrant Phone: +1.6506234000
Registrant Phone Ext:
Registrant Fax: +1.6506188571
Registrant Fax Ext:
Registrant Email: dns-admin@google.com
Registry Admin ID:
Admin Name: DNS Admin
Admin Organization: Google Inc.
Admin Street: 1600 Amphitheatre Parkway
Admin City: Mountain View
Admin State/Province: CA
```

Figure 3.10: Example WHOIS query report. Client search for WHOIS information of `ssl.gstatic.com` on the host `whois.markmonitor.com`. Registrant's websites can be extracted from "Registrant email" entry (highlighted in red rectangle).

contain a secondary level domain (SLD) besides TLD. For example, `www.bbc.co.uk` where *co* is the SLD. So our domain name filter should first determines the "length" of "(SLD+)TLD".

Furthermore, the WHOIS server of a domain name also depends on "(SLD+)TLD" especially those ending with country code TLD. For example, the WHOIS server of country code TLD *cn* is `whois.cnnic.net.cn`[8]. Therefore, given a domain name, we should first extract "(SLD+)TLD" then filter out the hostname to form a valid query and send request to corresponding WHOIS server based on "(SLD+)TLD". We use the WHOIS server list maintained by NirSoft [8] which contains the default *NIC WHOIS server of common TLDs as well as SLD+TLDs. The list works well in our project (78% of 2033 WHOIS queries success).

2. **Secondary Query**: The Network Information Center (NIC, as known as InterNIC) was once responsible for the allocation of all domain names. However, as stated in InterNIC's FAQ, domain names ending with .aero, .biz, .com, .coop, .info, .museum, .name, .net, .org, or .pro now can be registered through other different company (registrar)[9]. Therefore, we will need secondary query to redirect to

[8]Nir Sofer, WHOIS servers list for all domain types, `http://www.nirsoft.net/whois_servers_list.html`

[9]InterNIC FAQs on the Domain Names, Registrars, and Registration, https://www.internic.net/faqs/domain-names.html, retrieved on 12 August 2016

correct WHOIS server when we searching for domain names that not registered at InterNIC. In practice, querying domain names that not registered in its default NIC would return the registrar information of target domain name as well as the registrar information of other sub-domain of the target. Figure 3.11 illustrates the WHOIS report when searching for `google.com` on default NIC WHOIS server (WHOIS server of .com).

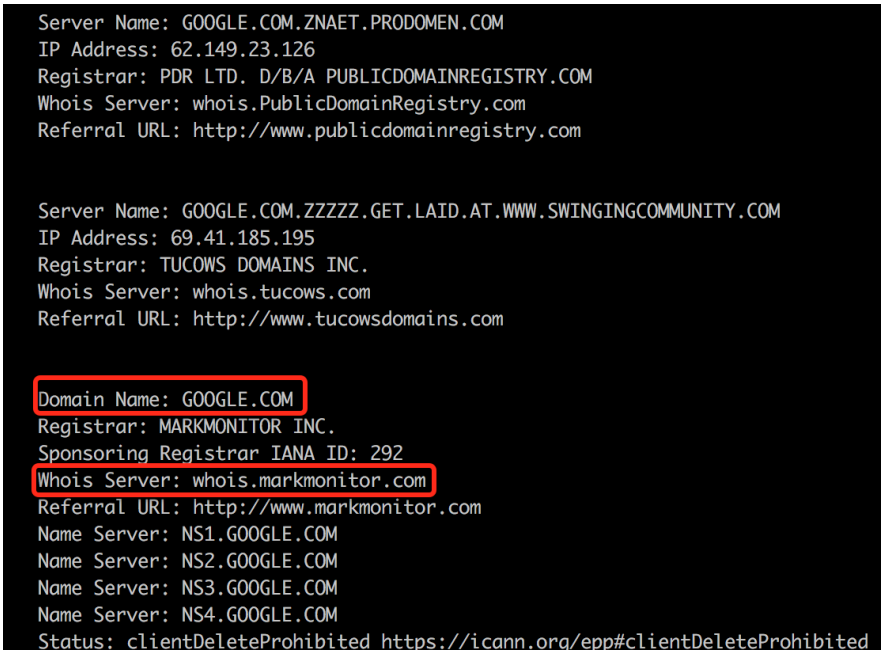


Figure 3.11: WHOIS report for querying google.com on .com default NIC WHOIS server[10]. The actual WHOIS server is the last identical entry on the report (highlighted in red rectangle).

Therefore, we can locate the actual WHOIS server address on the report via regular expression and query again on the correct WHOIS server.

3. **Report Format**: WHOIS service has its own drawback. The original RFC does not specify any file format or text template of the WHOIS report so different WHOIS server may use different report formats. Therefore, it is not feasible to create specific regular expression for every single WHOIS server. Fortunately, most third-party URL (93.24%, 3173 out of 3403) in our case end with *.com* and *.net* and corresponding WHOIS servers follow almost the same report format so we only need a few regular expressions to parse these report.

[10]Generated by Linux command: whois '=google.com', '=' refer to search for the exact result of the query.

To summarize, PPE aims to extract the URL of privacy policy page for the given websites. It work as follows:

1. Loads the given website in Firefox and attempts to identify the target link by its text.

2. If fail, queries WHOIS server of the given website: first query on default NIC WHOIS server then query on redirected WHOIS server if necessary, finally extract the URL of registrant's websites from WHOIS report.

3. Matching link text again on registrant's websites, if fail again, output NONE, output URL of privacy policy page otherwise.

3.4 Collecting Configuration and Results

We collected privacy policies from top 100 U.S. websites ranked by `Alex.com`[3]. Our collecting configurations are as follows:

- **Hardware**: 2.9GHz dual-core Intel Core i5, 8GB 1866MHz LPDDR3 memory, 512GB PCIe-based flash storage.

- **Operating System**: OS X EI Capitan 10.11.6.

- **Software**: Firefox 46.0.1.

- **Runtime**: Java 1.8.0_91.

- **Collecting Time Period**: 30 June 2016 to 18 July 2016.

3.4.1 Collecting Result

We first run PPG to collect the links of privacy policy page then visit each of them manually to extract the text of privacy policy. For the top 100 U.S. websites (TOP100), our privacy policy gather (PPG) successfully collected 2560 privacy policy links from total 3403 contacting websites (75% success rate). 63.4% of the successfully links are extracted by querying WHOIS (see Figure 3.12).

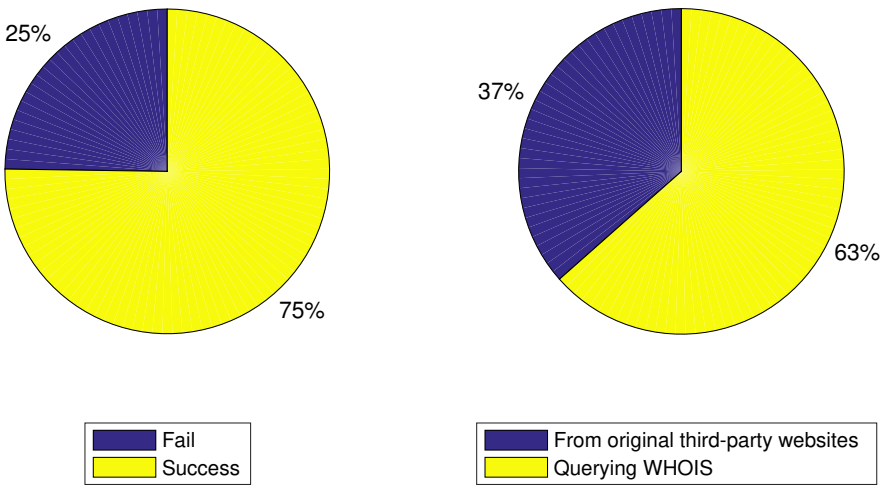


Figure 3.12: Success Extracting Rate and Corresponding Collecting Method. The left pie chart indicates 75% of total 3403 links are successfully extracted. The right pie chart shows that 63% of the successful links are extracted by querying WHOIS.

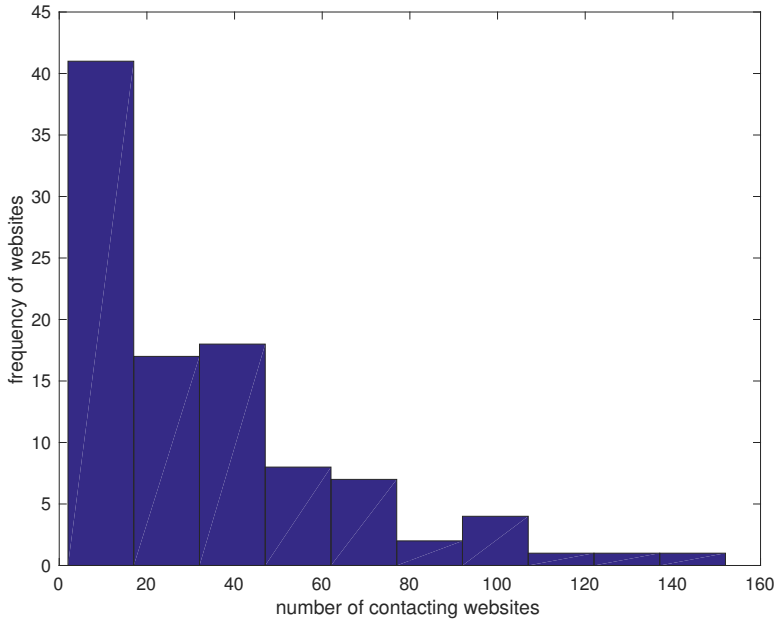


Figure 3.13: Histogram of number of contacting websites. 41% first-party websites contact less than 20 third-party websites (leftmost bin).

3.4.2 Collecting Result and Post-process

Figure 3.13 illustrates the histogram of the number of contacting websites for each TOP100 websites. 41% of them contact less than 20 websites.

We then used fuzzy search in SQL to divided the contacting websites into first and third-party websites. At this moment, we consider a websites is third-party if its

URL (in the form of "domain+TLD") is different from TOP100 websites. Figure 3.2 demonstrates two contacting websites of first-party websites `yahoo.com` and its class under this division. Figure 3.14 illustrates the histogram of the ratio of third-party to all contacting websites and we found that third-party websites are contacted excessively among TOP100 websites.

Table 3.2: Example contacting websites of `yahoo.com`

TOP100 website	Contacting website	Class
yahoo.com	https://pixel.tapad.com	third-party
yahoo.com	https://ads.yahoo.com	first-party

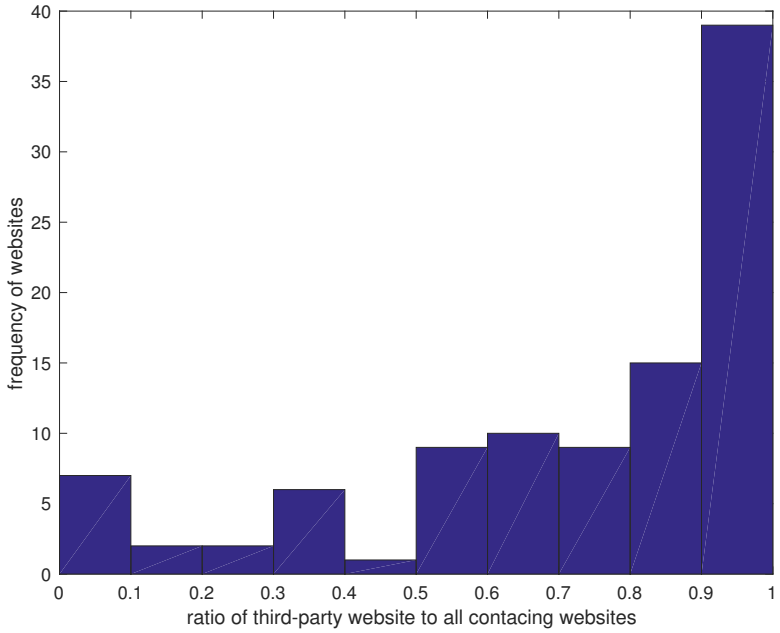


Figure 3.14: Histogram of the ratio of third-party to all contacting websites. 72% of TOP100 websites contacting third-party websites more than first-party websites (bins greater than 0.5).

After division, we obtained 205 privacy policy links and we visited each of them by hand to exclude duplicated links. We finally obtained 192 distinct privacy policy links[11], which included 79 from first-party[12] and 113 from third-party.

[11] Different websites may be covered by the same privacy policy provided by its parent company, for example, `youtube.comdoubleclick.net`, `gstatic.com` are both covered by `google.com`'s privacy policy.

[12] Some first-party websites failed as well

As we stated in Chapter 3, our PPG makes best effort to find the link of privacy policy but not guarantee the exact result since the location and format of privacy policy pages vary from website to website. We therefore extracted the text of privacy policy manually for these 192 links. To be specific, for each distinct privacy policy links, we find the correct privacy policy page if necessary and save the text content.

For each privacy policy, we prepared two format for later analysis. A full-text format which contains the full text for measuring usability and a structural format (stored in XML file) which preserves the paragraph structure of privacy policy for measuring topic similarity on paragraph level. A privacy policy paragraph is the top-most level section of the privacy policy defined by the websites and concerning about the one privacy topic (e.g., what personal information will be collected). Table 3.3 demonstrates two paragraphs of `yahoo.com` and `amazon.com`'s privacy policy. If we

Table 3.3: Example privacy policy paragraphs from `yahoo.com` (left) and `amazon.com` (right). The titles of each paragraph are shown in bold.

Yahoo Privacy Centre	**Amazon Privacy Policy**
...	...
Information collection & use	**What Personal Information About Customers Does Amazon.com Gather?**
Yahoo (as data controller) collects personally identifiable information when you register for a Yahoo account, when you use certain Yahoo products or services, when you enter promotions or competitions and when you visit Yahoo pages or the pages of certain Yahoo partners outside the branded Yahoo network of websites.	The information we learn from customers helps us personalize and continually improve your Amazon experience. Here are the types of information we gather.
...	...
Information sharing & disclosure	**Does Amazon.com Share the Information It Receives?**
Yahoo does not rent, sell or share information about you (including personally identifiable information) with other people or non-affiliated companies, except to provide products or services that you have requested, when we have your permission or in the following circumstances:	Information about our customers is an important part of our business, and we are not in the business of selling it to others. We share customer information only as described below and with subsidiaries Amazon.com, Inc. controls that either are subject to this Privacy Notice or follow practices at least as protective as those described in this Privacy Notice.
...	...

are able to identify the title of each paragraphs, it will be straightforward to divide the document into paragraphs. However, although we extracted the privacy policy and preserve its paragraph manually, we found this task is still non-trivial. As shown in Table 3.3, Even two paragraphs are concerning about the same privacy topic, the paragraph titles still vary from site to site (e.g., for information sharing, "Information sharing & disclosure" in `yahoo.com` comparing to "Does Amazon.com Share the Information It Receives" in `amazon.com`). This requires carefully manual review. Due to the time stress, we are not able to review all the privacy policies we have collected.

Instead, as a proof-of-concept, we randomly choose 6 websites from TOP100 and review all the privacy policies related to this 6 websites[13]. Figure 3.15 demonstrates one section of `yahoo.com`'s privacy policy (in structural format).

```
<POLICY id="184" policyUrl="https://policies.yahoo.com/us/en/yahoo/privacy/index.htm"
retriveDate="2016-07-01">
  <SECTION>
    <SUBTITLE>Yahoo Privacy Center</SUBTITLE>
    <SUBTEXT>Welcome to the Yahoo Privacy Center -- take a look around. You'll learn how
    Yahoo treats your
personal information, along with ways to control your preferences and settings. As always,
Yahoo is
committed to gaining your trust.</SUBTEXT>
  </SECTION>
```

Figure 3.15: A section of `yahoo.com`'s privacy policy after manual reviewing. Each section is a privacy policy paragraph. The title and content of each paragraph are wrapped in different XML element for later retrieve (*SUBTITLE* for title and *SUBTEXT* for content).

On the contras, saving the full text of each privacy policies is trivial so we managed to collect all 192 privacy policy in full-text version.

[13] 27 privacy policies in total

Chapter 4

Comparing Privacy Policies

This chapter describes how we compare the privacy policies we collected. We focus on counting sensitives words and comparing topic similarity via natural language processing (NLP) and measuring the usability of privacy policies (readability and cost of time to read).

4.1 Comparing Privacy Policy via NLP

The following sections illustrate how we compare collected privacy policy via NLP. We used NLTK[1] and Gensim[2] to process our data. The former provides rich APIs to perform NLP in Python and the later is a Python library dedicates to topic modeling..

4.1.1 Preprocess

We first transformed the raw privacy policies into corpus for later NLP process. These preprocesses are both operated on full-text and structural format:

1. **Tokenization**: Since all the collected privacy policies are in English, each word separated by space is a *term*. We therefore splited every sentence by space in the collection to generate terms.

2. **Excluding Stop Words**: Stop words are those most common words in a language (e.g., "the" in English) and are excluded before processing. Those words occurs frequently while appear be to be less valuable to understand a document (Manning et al., 2008) We used the English stop word list provided by Standford NLP

[1] Natural Language Toolkit (NLTK), `http://www.nltk.org`
[2] Gensim, `https://radimrehurek.com/gensim/`

Group[3] which contains punctuations and most common English stop words. We also excluded the URL of website itself and website's name for each website's privacy policy (e.g., `amazon.com` and "Amazon" for `amazon.com`'s privacy policy) since these words also occur frequently in privacy policies but appear to be less important to us.

3. **Stemming**: Since we were not interested in the form of the words, we will stem the text in the preprocess. Stemming aims to reduce the inflectional and morphologically derived form of a term to a common base form (Manning et al., 2008). For example, reduce *shared*, *sharing* to *share*. Therefore, if *shared*, *sharing* and *share* all appear once in a document respectively, the frequency of the term *share* will be 3. We used the Porter stemmer, which has been approved to be empirically effective (Harman, 1991), for this task.

4.1.2 Counting Sensitives Words

After having a discussion with a NLP expert from the faculty who does research on negation, we confirmed that finding the exact negation in natural language is not viable. However, she suggested that counting the frequency of sensitive words in privacy policy can be a reasonable start point. To be specific, a company is more likely to repeat a word many times if the company actually uses it while states less if it would not use it. We followed this idea and chose *share* and *disclose* as our target since these two words were strongly related to user's privacy. We used the full-text format of privacy policy for this task.

We counted the frequency of *share* and *disclose* in 192 privacy policies. We drew the histogram (see Figure 4.1) and box plot (see Figure 4.2) of the frequency of *share*, *disclose* on both first and third-party websites' privacy policies, respectively.

As shown in Figure 4.2, all the box skew to bottom indicating that most first and third-party websites appear to use these two word less frequently, specifically, less than 5 times (see Figure 4.1). Furthermore, in Figure 4.2, *share_f* has a large variability than other three boxes, meaning that using *share* varies more considerably from website to website on first-party websites.

To verify the initial idea mentioned at the beginning of this subsection, We then went through those privacy policies using *share* less then once (including once) manually

[3]The Standford Natural Language Processing Group, `https://github.com/stanfordnlp/CoreNLP/blob/master/data/edu/stanford/nlp/patterns/surface/stopwords.txt`, retrieved on 24 June 2016

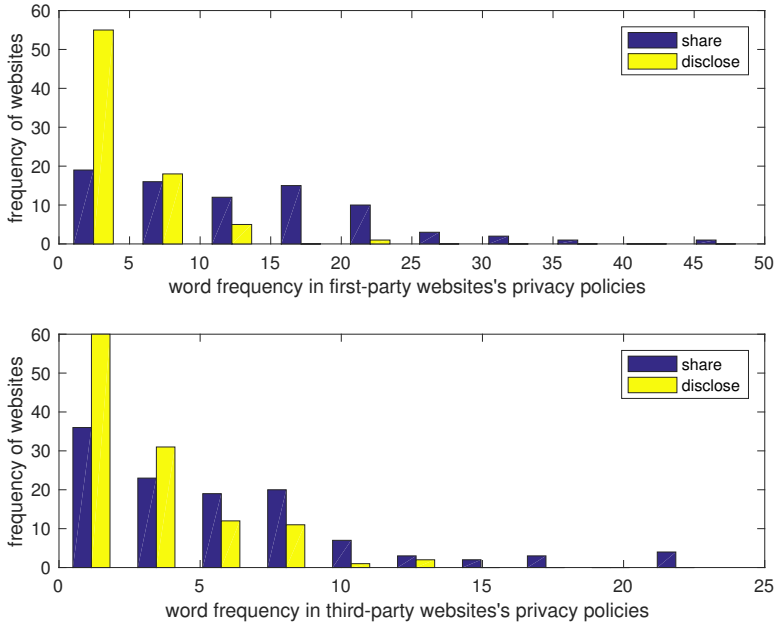


Figure 4.1: Histogram of the frequency of *share* and *disclose* in first (upper) and third-party (lower) websites' privacy policies.

since these privacy policies will probably not share personal information under the initial idea. Manual examination is to determine whether a privacy policy will or will not share the personal information. This statement is clear to be identify in a privacy policy. We considered the following criteria to determine whether a websites will share or not personal information (see Table 4.1):

Table 4.1: Criteria to determine whether a websites share or not personal information

statement	share/not share
not collect personal information	not share
not share/pass/rent/sell/disclose personal information	not share
only share non-personal information	not share
only use personal information for this website's service	not share
only share personal information if required by law	not share
will share/pass/rent/sell/disclose personal information	share
not share/pass/rent/sell/disclose personal information except following circumstances	share
provide personal information with advertiser and publisher	share

We found this solution does not work well. Among 34 privacy policy samples

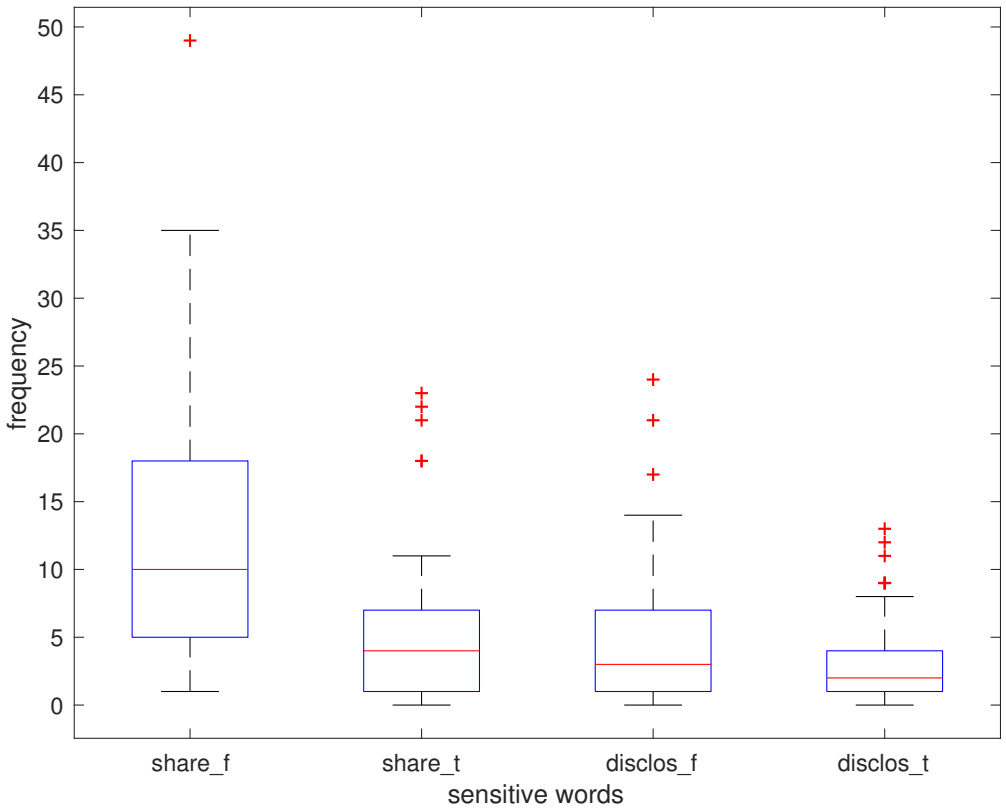


Figure 4.2: Box plot of the frequency of *share* and *disclose* in first-party (*share_f* and *disclose_f*) and third-party (*share_t* and *disclose_t*) websites' privacy policies. In each box, the horizontal lines refer to maximum, third quartile, median (in red), first quartile and minimum from top to bottom, respectively. Outliers are shown in red plus.

which use *share* less than once (including once), only 31% stated will not share collected personal information. However, this conclusion need to be verify on larger dataset and may also require the help from law and privacy policy experts to examine the aforementioned criteria.

4.1.3 Topic Similarity

To compare topic similarity, we used the structural format privacy policies. Each policy comprises of many sections and each section has its section title and content. A section is a privacy policy paragraph namely, the top most section in the original privacy policy. Table 4.2 demonstrates two paragraphs in the original privacy policy and the corresponding sections in structural format

After transforming original privacy policy into structural format, we can easily access each element in a document. We then build corpus for each unique privacy

Table 4.2: Original privacy policy of `yahoo.com` and corresponding structural format policy

Original privacy policy	**Structural format**
	...
	<**SECTION**>
	<**SUBTITLE**>**Information collection & use**
	<**/SUBTITLE**>
	<**SUBTEXT**>
...	Yahoo (as data controller) collects personally identifiable
Information collection & use	information when you register for a Yahoo account
Yahoo (as data controller) collects personally identifiable information when you register for a Yahoo account	<**/SUBTEXT**>
	<**/SECTION**>
...	...
Information sharing & disclosure	<**SECTION**>
Yahoo does not rent, sell or share information about you (including personally identifiable information) with other people or non-affiliated companies	<**SUBTITLE**>**Information sharing & disclosure**
	<**/SUBTITLE**>
	<**SUBTEXT**>
...	Yahoo does not rent, sell or share information about you (including personally identifiable information) with other people or non-affiliated companies
	<**/SUBTEXT**>
	<**/SECTION**>
	...

policy from this format's file, specifically, build the term-document matrix (TDM) for each unique privacy policy. Here a document is a section in structural format and all the sections of a privacy policy form its corpus. In addition, each section is a row in TDM.

For each first-party websites' privacy policy, we computed the topic similarity score (TSS) between it and every contacting third-party website's privacy policy on paragraphs level. The basic idea is to compute the distance between two paragraph (one from first-party another one from third-party) in the topic space. Explained as follows:

From the perspective of LDA, a document (in the form of bag-of-words) is a distribution of topics. So assigning a topic to a document can be considered as a process of dimension reduction, in other words, projecting TDM into topic space which has less number of dimensions (Blei et al., 2003). Then we can compute the distance between two projected vector in topic space as a metric of topic similarity between two document. Smaller the distance, more similar the two document.

Assume we have the TDM of a first-part websites' privacy policy m_f and TDM of one of its contacting third-party websites' privacy policy m_t, we first run LDA on

m_f and m_t respectively and obtain two LDA model M_f and M_t. To compute topic similarity under M_f, first assgin a topic under M_f to each section of m_f and m_t so we have two projected matrices m_{ff} and m_{tf}. For each project document in m_{tf} (a row vector), compute the distance to each section of m_{ff} and find the maximum. If the maximum still less than a threshold, this section is considered to be dissimilar to any section in m_f and we increment a counter n'_f for the number of dissimilar sections in m_t under M_f. Then compute topic similarity under M_t as the same procedure above but count the number of dissimilar sections in m_f in counter n'_t. Finally, the topic similarity score for m_f and m_t is defined as follows (Equation (4.1)):

$$tss = 1 - \frac{n'_f + n'_t}{n_f + n_t} \tag{4.1}$$

tss: the similarity score

n_f: number of sections in m_f

n_t: number of sections in m_t

Higher the score, more similar the two privacy policy in terms of topics. The above procedure is summarized in Algorithm 1.

Algorithm 1 Compute topic similarity

Input: TDM of first-party websites' privacy policy m_f, TDM of contacting third-party websites' privacy policies m_t, number of topic to generated n_t and similarity threshold $treshold$

Output: topic similarity score tss

function TOPICMODELING(TDM m, n_t)
    Run LDA on m, generate LDA model M and n_t topics $topic$
    return M
sec_f = number of sections in m_f ▷ number of rows
sec_t = number of sections in m_t ▷ number of rows
$sec'_f = 0$, $sec'_t = 0$
$m_{ff} = []$, $m_{tf} = []$, m_{ft}=[], m_{tt}=[]
M_f = TOPICMODELING(m_f, n_t)
M_t = TOPICMODELING(m_t, n_t)
▷ project m_f into topic space
for sec in m_f **do**
    m_{ff}[sec] = topic for sec under M_f
    m_{ft}[sec] = topic for sec under M_t
▷ project m_t into topic space
for sec in m_t **do**
    m_{tf}[sec] = topic for sec under M_f
    m_{tt}[sec] = topic for sec under M_t
for sec in m_{tf} **do**
    $D = []$
    for sec' in m_{ff} **do**
        D[sec'] = distance between $m_{ff}[sec]$ and $m_{tf}[sec']$
    if MAX(D)< $threshold$ **then**
        $sec'_t += 1$
for sec in m_{ft} **do**
    $D = []$
    for sec' in m_{tt} **do**
        D[sec'] = distance between $m_{ft}[sec]$ and $m_{tt}[sec']$
    if MAX(D)< $threshold$ **then**
        $sec'_f += 1$
Output: $tss = 1 - \frac{sec'_f + sec'_t}{sec_f + sec_t}$

We ran this algorithm on 6 TOP100 websites' privacy policies and their contacting third-party websites' privacy policies (27 privacy policies in total) and chose to generate 7 topics per policy. If we choose more topics to generate, LDA will produce many duplicated topics which is unexpected (see Appendix A).

Figure 4.3 illustrates an example result for `gfycat` and its contacting third-party websites' privacy policies. To evaluate the generated topics, we first visualized the

id	T->F	F->T	Total paragraphs	TSS
https://gfycat.com/privacy	-1	-1	13	-1
https://www.quantcast.com/privacy/	7	3	32	0.7777777777777778
https://www.amazon.com/gp/help/customer/…	3	6	12	0.64
https://twitter.com/privacy	2	6	6	0.5789473684210527
https://www.facebook.com/privacy/explanation	3	9	9	0.4545454545454546
https://www.google.com/intl/en/policies/privacy/	5	6	14	0.5925925925925926
http://www.alexa.com/help/privacy	2	2	16	0.8620689655172413

Figure 4.3: Example topic similarity score of `gfycat` and its contacting third-party websites' privacy policies. "F->T" means the number of dissimilar sections in third-party websites' privacy policy (sec'_f). Similarly, "T->F" means that the dissimilar sections in first-part websites' privacy policy (sec'_t). First record refers to first-party websites' privacy policy, as reference.

frequency of top 50 topic words in word cloud (see Figure 4.4). As expected, many

Figure 4.4: Word cloud of top 50 topic words. The word with higher frequency is represented in larger font size.

privacy related words occurs more frequently in the generated topics[4] (e.g., information, use, privacy, advertise). By plotting the histogram of topic similarity scores, we found that this scores is distributed normally: centering more in the middle and less in head and tail (see Figure 4.5). Furthermore, the scores vary from 0.08 to 0.91, extremely dissimilar to considerably similar in terms of topics.

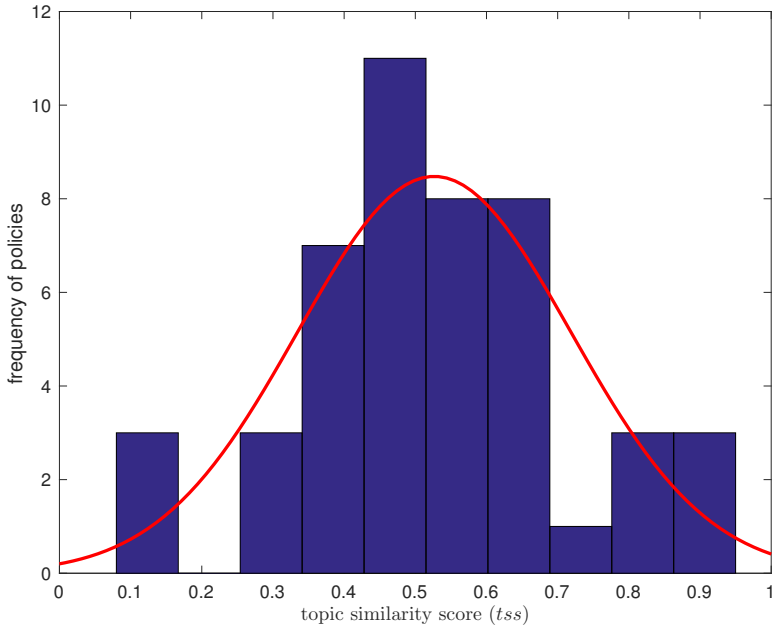


Figure 4.5: Histogram of the topic similarity score in testing websites' privacy policies. The red curve is the normal distribution fitted by the mean and covariance of the frequency of TSS.

4.2 Measuring Usability of Privacy Policy

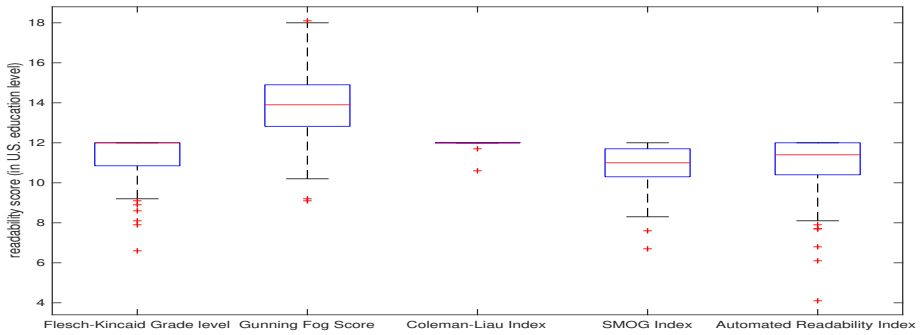
This section describes how we measure the usability of collected privacy policies, i.e., readability and cost of time to read. We also present our findings of question mentioned in Chapter 2: assume a user knows there are many third-party websites are connecting with site he initially visits, how much time will it cost to read all these privacy policies?

4.2.1 Readability

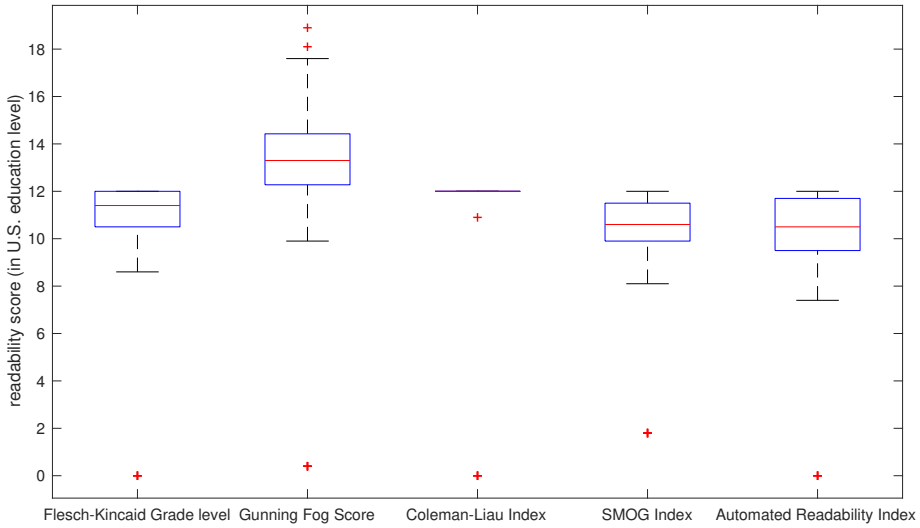
We calculated the readability score of 192 privacy policies using several readability formula: Automated Readability Index, ColemanLiau Grade, FleschKincaid Grade,

[4] see Appendix B

Flesch Reading Ease, Gunning Fog Grade, SMOG Grade. We used full-text format privacy policies and PHP Text-Statistics library to compute the scores[5]. Figure 4.6



(a) Readability scores of first-party websites' privacy policies



(b) Readability scores of first-party websites' privacy policies

Figure 4.6: Box plots of the readability scores of first (upper) and third-party websites' privacy policies

As shown in Figure 4.6, all the privacy policies require at most 12 education level to read under four metrics, which is the senior high school level in U.S. Gunning Fog Score produces a considerably higher readability score, indicating harder to read, than other metrics. The most complex privacy policy even require postgraduate education level (18 in the figure) under Gunning Fog Score. In addition, there is not obvious differences between first and third-party websites' privacy policy in terms of readability.

[5]PHP Text Statistics, `https://github.com/DaveChild/Text-Statistics`

4.2.2 Cost of Time to Read

McDonald and Cranor (2008) estimated the annual time to read privacy policies for individuals and the U.S. nation based on privacy policies from 75 most popular websites. Their study did not address on the presence of third-party websites. We reproduced their estimation when a user need to read the privacy policies from first-party website he initially visiting as well as the contacting third-party websites.

The estimation including time to read entire privacy policy and time to skim. The following formulas defines the estimated annual time to read ("T_R") and skim ("T_S")privacy policies (McDonald and Cranor, 2008):

$$T_R = p * R * n \tag{4.2a}$$

$$T_S = p * S * n \tag{4.2b}$$

p: the population of Internet users

R: the average time to read a entire privacy policy

S: the average time to skim a privacy policy

n: the average number of unique sites an Internet user visits each year To estimate R, we first computed the word length for 192 privacy policies then timed 250 WPM[6]. Figure 4.7 demonstrates the histogram of word length. We found that the word length of 192 collected privacy policies varies from 282 to 19010. Then we chose first quartile, median and third quartile of the word length as the estimated point and divided each of them by 250WPM. Table 4.3 illustrates this result:

Table 4.3: Estimated time to read a privacy policy

	Word length		Average reading rate	Time to read one privacy policy
short policy (first quartile)	1628	/	250WPM	=6.5
medium policy (median)	2600	/	250WPM	=10.4
long policy (third quartile)	3254	/	250WPM	=13

To estimate T_S, we used the data stated in (McDonald and Cranor, 2008). Their estimated T_S was obtained via online survey which required participants to skim a

[6]Typical reading rate for people with high school education (McDonald and Cranor, 2008)

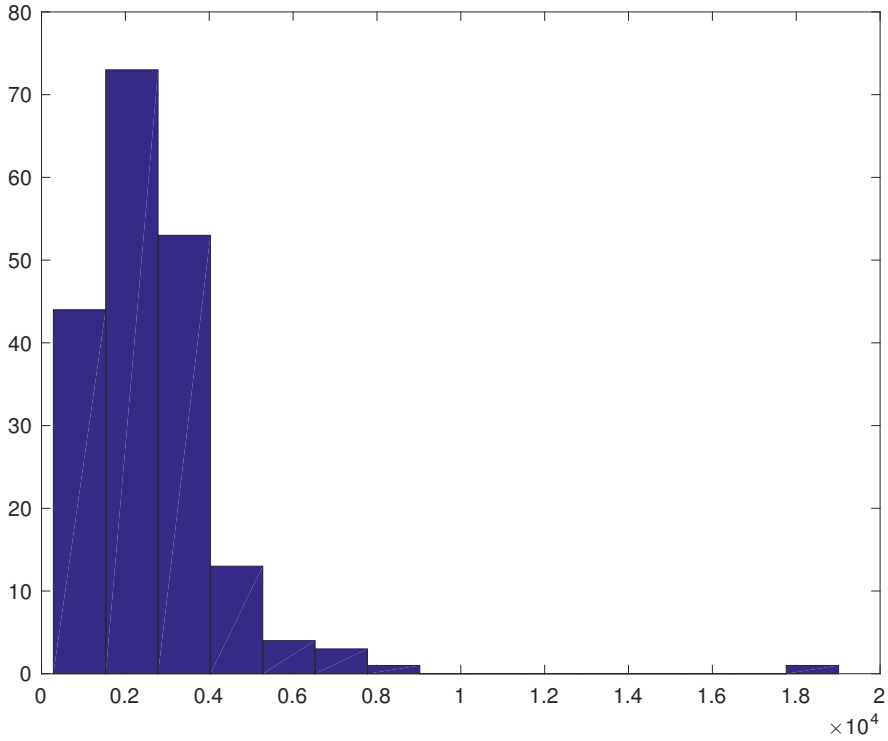


Figure 4.7: Histogram of the word length on collected privacy policies.

privacy policy and answered one basic question, For example, when is the last updated time of the policy? (McDonald and Cranor, 2008). The result is shown in Table 4.4:

Table 4.4: Estimated time to skim a privacy policy

	Time to skim a privacy policy and answer one basic question
low estimate	3.6min
point estimate	6.3min
high estimate	11.6min

In McDonald and Cranor's study, they estimated n based on the monthly number of unique websites a user visits reported by Nielson Online[7]. When considering the presence of third-party websites, we estimated n as follows:

$$n = n_u * n_t$$

n is the annual number of unique websites a user visits with the presence of third-

[7]Unfortunately, the updated reports require charge to access so we kept McDonald and Cranor's data.

party websites.

n_u is the annual number of unique first-party websites a user visits.

n_t is the number of unique websites a user need to visits when he initially visits one first-party websites (i.e., first-party websites and contacting third-party websites).

In our case, $n_t = 1.92$ since we finally obtained 192 unique privacy policies after visiting 100 first-party websites. The result of the estimate annual unique websites a user visits with the presence of third-party websites are summarized in Table 4.5:

Table 4.5: Estimated annual number of unique websites (n) a user visits with the presence of third-party websites

	n_u	n_t	n
low estimate	1354/ year	1.92 per first-party website	2600/ year
point estimate	1462/ year	1.92 per first-party website	2807/ year
high estimate	1518/ year	1.92 per first-party website	2915/ year

Finally, we multiply the annual number of unique websites a user visits with the presence of third-party websites (n) by average time to read or skim a privacy policy, and by estimated population of American Internet users (283,712,407[8]). The result is shown in Table 4.6:

Table 4.6: Estimated annual time of reading or skimming privacy policies with the presence of third-party websites

	Individual time to read (hours / year)	Individual time to skim (hours / year)	National time to read (hours / year)	National time to skim (hours / year)
low estimate	282	156	80 billion	44 billion
point estimate	487	295	138 billion	83 billion
high estimate	632	563	179 billion	159 billion

Even with low estimate, a user still need spend 282 hours reading the privacy policies of the website he visits with the presence of third-party websites.

[8]United States Internet Users, Internet Live Stats, http://www.internetlivestats.com/internet-users/us/, retrieved on 25 July 2016

Chapter 5

Discussion

In this chapter, we describe the lessons we learnt from collecting, comparing and measuring privacy policies.

5.1 Collecting Privacy Policy

From the procedure of collecting privacy policies from first-party websites and contacting third-party websites (see Chapter 3), we found that querying WHOIS is a reasonable solution to find the privacy policies of those human unreadable websites. Those websites are designed for exchanging data between web servers and are common to be seen in the list of contacting websites in network monitor.

Finding specific web pages across many websites is still problematic for machine and relies heavily on human's assistance. For example, the privacy policy of a websites can across many web pages, buried deep within or even mingled with "Terms of Use" (Liu et al., 2014). Even have found the target pages, extracting the text content still require manual process since the privacy policy pages can be in PDF format or in various of HTML structures among the websites.

5.2 Comparing Privacy Policy: Topic Similarity

Automatically parsing privacy policies is not feasible, even manual parsing is still non-trivial since privacy policy is excessive in length and difficult to read Luger et al. (2013). As a proof-of-concept, we parsed 27 privacy policies and preserved the text structure to compute the topic similarity.

By counting the word frequency, we did not see a strong relation between word frequency and negation: only 31% of privacy policies that use *share* less than once clearly stated they will *not share* personal information.

We defined topic similarity score to measure two privacy policy in term of topics and computed TSS on our sample policies. The frequency of the TSS score distributed similarly to normal distribution. We are longing to obtain more samples to see whether this distribution is a general trend on the web.

5.3 Measuring Usability of Privacy Policy: Readability and Cost of Time to Read

By measuring the readability of collecting privacy policies, we noticed that three metrics reported at most grade 12 education level was needed to read the privacy policies while one reported a larger range of necessary education levels: varies from 5 to 18 which is the postgraduate level. These differences between readability metrics is probably depends on how they define the syllables.

From the result of measuring cost of time to read privacy policies, a user need to read 1.92 policies per websites he visits with the presence of third-party websites and such reading will cost him 487 hours a year and 83 billion hours for the U.S. nation. This statics is still underestimated since we ignored the time to find the privacy policies of third-party websites which is also a non-trivial task as stated in Section 5.1.

Chapter 6

Conclusion and Future Work

We mainly contributed to build a semi-automatic privacy policy gather (PPG), develop an algorithm to compare two privacy policies in terms of topic similarity and measure the usability of the privacy policies (readability and cost of time to read). By using PPG, we have successfully collected 192 unique privacy policies from 3403 connections generated by visiting top 100 U.S. websites and the success rate of PPG is 75.22% (see Chapter 3. For comparing privacy policies, we defined an equation to evaluate the topic similarity (topic similarity score, see Section 4.1.3) between two privacy policies and applied the algorithm to 27 policy samples as a proof-of-concept. We found the frequency of topic similarity score distributed similarly to normal distribution and varied from 0.08 to 0.91, extremely dissimilar to considerably similar in terms of topics. For those extremely dissimilar policies, more attention need to be paid to identifying potential conflicting privacy statements.

By measuring the readability of collected privacy policies, we found three metrics agreed on at most 12 education level is required to read privacy policies while one metrics reported a education level varied from 8 to 18, which is the postgraduate level. Moreover, based on McDonald and Cranor's study, we measured the cost of time to read privacy policies with the presence of third-party websites. We a user need to read 1.92 privacy policies per website he visits and it will cost him 487 hours a year and 83 billion hours for the U.S. nation. The results from the measurement of readability and cost of time indicate more improvement can be made to alleviate the burden to read privacy policies.

In the future, we hope to examine the topic similarity score and the relation between word importance and negation on larger data set. CMU's Usable Privacy Policy[1]

[1]The Usable Privacy Policy Project, `https://usableprivacy.org`

utilized crowdsourcing to parsing 115 privacy policies. Each sentences is classified into several privacy statements categories by three crowd workers. This highly structural and tagged data can not only benefit measuring topic similarity using unsupervised topic modeling but also provide potential to run supervised machine learning algorithms.

Bibliography

Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM*, 55(4):77–84.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Federal Trade Commission (2013). Children's online privacy protection rule. `https://www.ftc.gov/system/files/2012-31341.pdf`. Online; accessed 4 August 2016.

Graber, M. A., Johnson-West, J., et al. (2002). Reading level of privacy policies on internet health web sites.(brief report). *Journal of Family Practice*, 51(7):642–646.

Harman, D. (1991). How effective is suffixing? *Journal of the American Society for Information Science*, 42(1):7.

Hochhauser, M. (2001). Lost in the fine print: Readability of financial privacy notices. *Privacy Rights Clearinghouse*.

Hofmann, T. (1999). Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM.

Hoofnagle, C. J., Soltani, A., Good, N., Wambach, D. J., and Ayenson, M. D. (2012). Behavioral advertising: the offer you cannot refuse.

Jensen, C. and Potts, C. (2004). Privacy policies as decision-making tools: an evaluation of online privacy notices. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 471–478. ACM.

Liu, F., Ramanath, R., Sadeh, N., and Smith, N. A. (2014). A step towards usable privacy policy: Automatic alignment of privacy statements.

Luger, E., Moran, S., and Rodden, T. (2013). Consent for all: revealing the hidden complexity of terms and conditions. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 2687–2696. ACM.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

McDonald, A. M. and Cranor, L. F. (2008). Cost of reading privacy policies, the. *ISJLP*, 4:543.

McDonald, A. M., Reeder, R. W., Kelley, P. G., and Cranor, L. F. (2009). A comparative study of online privacy policies and formats. In *Privacy enhancing technologies*, pages 37–55. Springer.

Nikiforakis, N., Kapravelos, A., Joosen, W., Kruegel, C., Piessens, F., and Vigna, G. (2013). Cookieless monster: Exploring the ecosystem of web-based device fingerprinting. In *Security and privacy (SP), 2013 IEEE symposium on*, pages 541–555. IEEE.

Papadimitriou, C. H., Tamaki, H., Raghavan, P., and Vempala, S. (1998). Latent semantic indexing: A probabilistic analysis. In *Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pages 159–168. ACM.

The European Parliament and The Council of The European Union (1995). Data protection directive 95/46/ec. `http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:en:HTML`. Online; accessed 4 August 2016.

United States Government Accountability Office (2013). Information resellers consumer privacy framework needs to reflect changes in technology and the marketplace. Technical Report GAO-13-663, United States Government Accountability Office. `http://www.gao.gov/assets/660/658151.pdf`.

Appendix A

LDA with Different Number of Topic to Generate

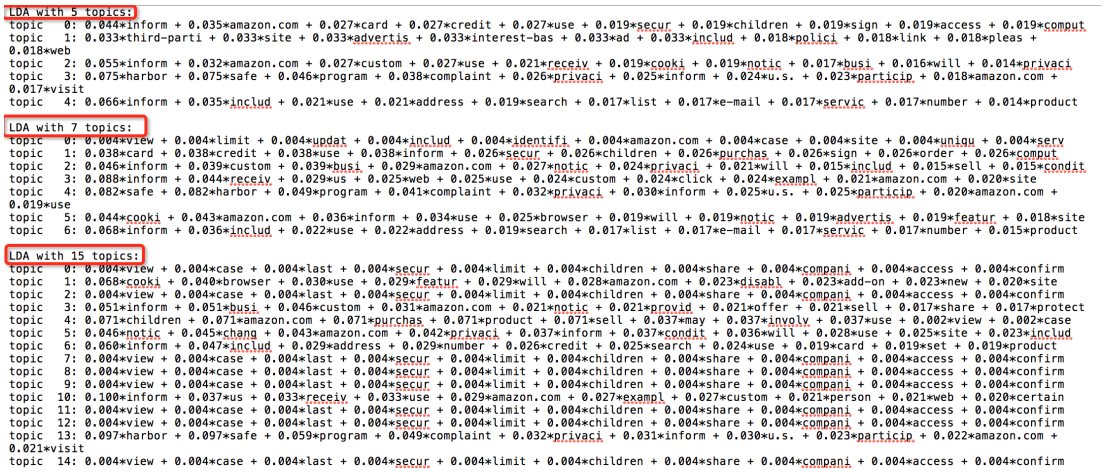


Figure A.1: Generated topics by LDA with different number of topic to generate. Choosing more topics to generate will produce many duplicated topics which is unexpected.

Appendix B

Frequency of Top 50 Topic Words

word	count	word	count	word	count	word	count	word	count
information	129	advertise	42	change	20	email	13	regard	9
use	96	website	34	ad	19	delete	13	purpose	9
may	80	will	31	share	18	time	13	product	9
service	67	include	31	account	17	company	12	web	9
privacy	61	contact	30	site	17	secure	12	post	9
provide	51	data	30	set	16	trust	12	address	9
collect	51	cooky	28	device	16	visit	10	yahoo	9
person	50	please	25	browser	16	question	10	party	9
us	46	identify	22	page	15	receive	10	program	8
policy	44	access	21	user	14	opt	10	direct	8

Figure B.1: The frequency of top 50 topic words